LWA 2009

Workshop-Woche: Lernen – Wissen – Adaptivität
Darmstadt, 21. - 23. September 2009

Editors
Melanie Hartmann und Frederik Janssen
Technische Universität Darmstadt

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Technical Report TUD-CS-2009-0157
FG Telekooperation, Technische Universität Darmstadt, Germany

Technical Report TUD-KE-2009-04
FG Knowledge Engineering, Technische Universität Darmstadt, Germany

Preface

The joint workshop event LWA 2009 (Lernen – Wissen – Adaptivität) was held from Sept., 21-23 2009 in Darmstadt, Germany. Like in the years before the LWA hosted four different workshops of the special interest groups of the Gesellschaft für Informatik (GI) for

- Adaptivity and User Modeling (FG-ABIS)
- Information Retrieval (FG-IR)
- Knowledge Discovery, Data Mining and Machine Learning (FG-KDML)
- Knowledge and Experience Management (FG-WM)

In addition to the separate workshops of each special interest group we invited two talks covering current research questions in computer science:

- *Relations and Probabilities: Friends, not foes*
  Kristian Kersting, STREAM group at Fraunhofer IAIS Bonn, Germany
- *Learning Valued Preference Structures: Toward an Alternative Decision-Theoretic Framework for Machine Learning*
  Eyke Hüllermeier, Knowledge Engineering & Bioinformatics Lab, Department of Mathematics and Computer Science at Marburg University, Germany

This years conference also featured a Tutorial on:

- *Mastering Unstructured Information with SMILA - the SeMantic Information Logistics Architecture*
  Igor Novakovic, empolis, Deputy Director Development

We are grateful for the support of several members of the Knowledge Engineering group and the Telekooperation group at the Technical University in Darmstadt, especially Gabriele Ploch for handling all the bureaucracy and our students George Ciordas, David Schuld, Florian Volk, Niklas Lochschmidt, Tim Klein, Raad Bahmani, and others for their technical support during the conference.

Melanie Hartmann and Frederik Janssen
Darmstadt, September 2009

We thank our sponsors for their generous support!

VEREINIGUNG VON FREUNDEN DER TECHNISCHEN UNIVERSITÄT ZU DARMSTADT E.V.

Capgemini sd&m

OdiSys GmbH

ABIS 2009

Workshop on Adaptivity and User Modeling in Interactive Systems

Editors

David Hauger, Johannes Kepler University Linz
Mirjam Köck, Johannes Kepler University Linz
Andreas Nauerz, IBM Research and Development

17th Workshop on Adaptivity and User Modeling in Interactive Systems (ABIS) 2009

David Hauger
Johannes Kepler University
Linz, Austria

Mirjam Köck
Johannes Kepler University
Linz, Austria

Andreas Nauerz
IBM Research and Development
Böblingen, Germany

The ABIS Workshop

ABIS - '*Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen*' - is the special interest group on Adaptivity and User Modeling in Interactive Systems of the German Computer Society.

The ABIS Workshop has been established as a highly interactive forum for discussing the state of the art in personalization and user modeling. Latest developments in industry and research are presented in plenary sessions, forums, and tutorials to discuss trends and experiences. The audience usually varies from young researchers working on Master or Ph.D. level to experts in the field. The workshop aims to provide a platform for exchanging novel ideas and expertise, and for obtaining feedback on ongoing research.

ABIS 2009

This year, we directly tied in with last year's ABIS which concentrated on the application of Web 2.0 and Social Computing technologies. In 2009, ABIS focused on exploiting ideas from the emerging fields of Web 3.0 and Ubiquitous Computing in order to come to even more sophisticated approaches for building user- and context models as well as improved mechanisms for adaptations and recommendations.

The program committee received submissions from research and industry, covering theoretical foundations but also discussing practical applications. Special emphasis was on submissions in the following areas:

- User- and context modeling for Web 2.0 / Web 3.0 / Ubiquitous Computing enabled adaptive systems (e.g. user behaviour analysis, data mining for personalization, community-based profiling)
- Web 3.0 / Semantic Web / Ubiquitous Computing technologies for adaptive systems (e.g. ontology-based user- and context models, reasoning on user- and context models)
- Adaptation techniques for Web 2.0 / Web 3.0 / Ubiquitous Computing enabled adaptive systems (e.g. tailoring information presentation to users)
- Recommender techniques for Web 2.0 / Web 3.0 / Ubiquitous Computing enabled adaptive systems (e.g. rule-based, knowledge-based, or content-based recommenders, leveraging collective intelligence, general aspects of collaborative filtering)
- Applications and case-studies of Web 2.0 / Web 3.0 / Ubiquitous Computing enabled adaptive systems (e.g. cost-justification of the application of these systems, convincing users of the added value)
- Methods for designing Web 2.0 / Web 3.0 / Ubiquitous Computing enabled adaptive systems (e.g. composition and management, evaluation, user studies)
- Behavior, social and cultural aspects of Web 2.0 / Web 3.0 / Ubiquitous Computing enabled adaptive systems (e.g. psychological and general usability aspects, privacy, trust and security)

Program committee

The program committee had the following members:

- Armen Aghasaryan, Alcatel-Lucent
- Mathias Bauer, mineway GmbH
- Manfred Broy, Technische Universität München
- Betsy van Dijk, University of Twente
- Michael Fahrmair, DoCoMo Euro-Labs
- Rosta Farzan, University of Pittsburgh
- Sabine Graf, National Central University
- Georg Groh, Technische Universität München
- Melanie Hartmann, Technische Universität Darmstadt
- Dominikus Heckmann, DFKI Saarbrücken
- Eelco Herder, L3S Research Center
- Jan Hidders, Technical University Delft
- Hagen Höpfner, International University Bruchsal
- Birgitta König-Ries, Universität Jena
- Arne Koesling, L3S Research Center
- Daniel Krause, L3S Research Center
- Tsvi Kuflik, University of Haifa
- Erwin Leonardi, Technical University Delft
- Stefanie Lindstaedt, Know-Center Graz
- Alexandros Paramythis, Johannes Kepler University
- Wassiou Sitou, Technische Universität München
- Kees van der Sluijs, Technical University Eindhoven
- Marcus Specht, Open University of the Netherlands
- Stephan Weibelzahl, National College of Ireland

We would like to thank the authors for their submissions, and we also thank the members of the program committee for providing helpful and constructive reviews.

September, 2009,

David Hauger, Mirjam Köck and Andreas Nauerz

ABIS

Table of Contents

Mashing up user data in the Grapple User Modeling Framework

Fabian Abel[1], Dominikus Heckmann[2], Eelco Herder[1], Jan Hidders[3], Geert-Jan Houben[3], Daniel Krause[1], Erwin Leonardi[3], Kees van der Slujis[4]
[1] L3S Research Center, Hannover, Germany, {abel,herder,krause}@L3S.de
[2] DFKI GmbH, Saarbrücken, Germany, heckmann@dfki.de
[3] WIS, TU Delft, The Netherlands{a.j.h.hidders,g.j.p.m.houben,e.leonardi}@tudelft.nl
[4] CS Department, Eindhoven University of Technology, The Netherlands, k.a.m.sluijs@tue.nl

Abstract

In this paper we demonstrate the Grapple User Modeling Framework (GUMF), which exploits Semantic Web technologies and Web 2.0 paradigms to model users across different applications and domains. It introduces novel features such as *dataspaces*, which logically bundle user data, and *user pipes*, which allow to mash up user data from different sources.

1 Introduction

Web systems such as Amazon or YouTube brought personalization to the general public. While those popular systems base recommendations on a large amount of data - by means of collaborative filtering and social network analysis - [Frias-Martinez *et al.*, 2005], the majority of Web applications cannot build upon a big user population and users might not interact regularly with these systems. A promising approach to compensate for this lack of data is *cross-application user modeling* [Korth and Plumbaum, 2007]. In contrast to the centralized approach of *generic user modeling servers* [Abel *et al.*, 2008], a *conversion-based* approach allows for flexible mappings. These mappings can be created from one system to another, or by making use of generalized representations, such as the General User Model Ontology (GUMO) [Heckmann *et al.*, 2005] and UserRDF [Abel *et al.*, 2008].

In this paper we present the architecture and implementation of the Grapple User Modeling Framework (GUMF) [Abel *et al.*, 2009a], which re-uses, refines and enhances previous work in the area of cross-application and generic user modeling systems. GUMF[1] organizes user profile data in *dataspaces*, which constitute *views* on a specific set of data. Dataspaces are extensible with plug-ins for mapping and integrating data from external data sources; these plug-ins can also be used to reason with existing to deduce further knowledge about the user. In addition to traditional rule-based approaches, GUMF provides so-called *user pipes* [Abel *et al.*, 2009a] that mash up different (user profile) data streams, formatted in RDF or RSS, making use of Semantic Web Pipes[2] or Yahoo Pipes[3].

2 GUMF: Grapple User Modeling Framework

Figure 1 shows the architecture of GUMF. The elements at the top provide the essential, generic functionality of the framework; elements part at the bottom right provide generic as well as domain-specific *reasoning logic*.

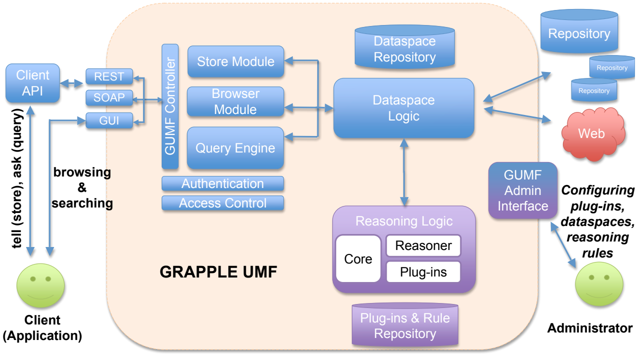


Figure 1: Architecture of the Grapple User Modeling Framework.

Client applications can access GUMF either via a RESTful or SOAP-based API. Further, there is a *Java Client API* that facilitates development of GUMF client applications. Client applications mainly approach GUMF to store user information (handled by the *Store Module*) or to query for information (handled by *Query Engine*). User profile information is modeled by Grapple statements [Abel *et al.*, 2009b], which are basically reified RDF statements about a user, enriched with provenance metadata. GUMF currently supports SPARQL and SeRQL queries as well as a pattern-based query language that exploits the Grapple statement structure to specify what kind of statements should be returned by GUMF. Authorized client requests are answered by GUMF's *Dataspace Logic*. Dataspaces are equipped with data storage repositories that either reside at the GUMF server or are distributed across the Web (possibly maintained by the client application itself), and with (reasoning) plug-ins that further enrich the data that is available in the repositories.

The *Administrator* of a GUMF client application can configure dataspaces and plug-ins via the *GUMF Admin Interface*. Activating or deactivating plug-ins directly influences the behavior of dataspaces. Further, administrators can adjust the plug-ins and reasoning rules to their needs. For example, we developed a plug-in that gathers user profile information from Facebook and maps—with support of

[1] Currently available at: `http://semweb.kbs.uni-hannover.de:8082/grapple-umf/`

[2] http://pipes.deri.org

[3] http://pipes.yahoo.com

Silk[4]—the profile to a format preferred by the client application administrator (e.g., FOAF[5] or OpenSocial[6]).

Inspired by Web 2.0 practices, a key principle of GUMF is that dataspaces can be shared across different client applications. Therefore, clients can *subscribe* to other dataspaces, given that they are granted approval by the administrator of the dataspace. When subscribed to a dataspace, the client is allowed to query it. However, it might still not be allowed to access all statements that are made available via the dataspace, as fine-grained access control functionality can be embedded in the dataspaces as well.

3 Demonstration Overview

In our demonstration we primarily show how client applications can benefit from the Grapple User Modeling Framework.

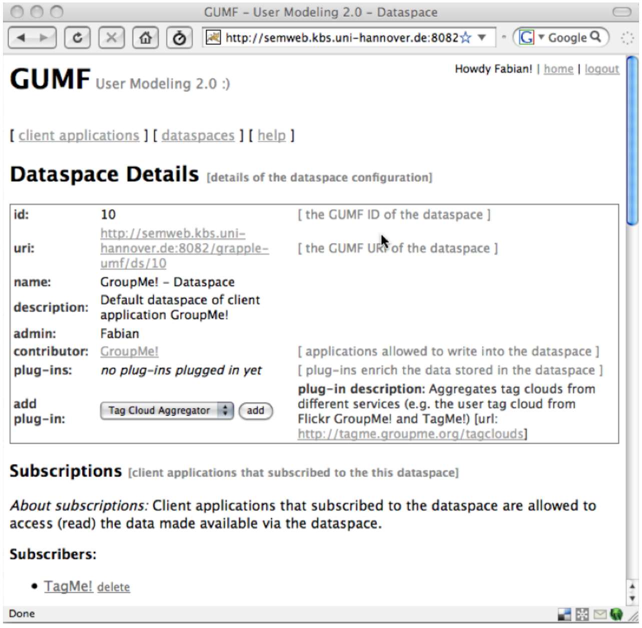
Developers of client applications first have to register their application at GUMF. Upon registration, a dataspace is generated that can immediately be used to store user profile information. As an example, a client might store user interests such as *"Peter is interested in Darmstadt"* in GUMF, by using the RESTful API and the Java Client implementation. GUMF models such information as Grapple statement.

```
<gc:Statement rdf:about="&ds10;6357701291243375806816">
  <gc:subject rdf:resource="&guser;peter"/>
  <gc:predicate rdf:resource="&foaf;interest"/>
  <gc:object rdf:resource="&dbpedia;Darmstadt"/>
  <gc:level rdf:datatype="&xsd;double">0.7</gc:level>
  <gc:origin>[peter(Interest: Darmstadt, 0.7]</gc:origin>
  <gc:created rdf:datatype="&xsd;dateTime">
    2009-05-27T00:10:06.817+02:00</gc:created>
  <gc:creator rdf:resource="&gclient;10"/>
</gc:Statement>
```

The core part of the Grapple statement consists of a subject-predicate-object triple, possibly extended with `gc:level` that describes to which degree the statement is true. In addition, the client can store the information in its original format (`gc:origin`). GUMF enriches the statement with metadata such as a globally unique ID, a timestamp (`gc:created`, which is a subproperty of `dc:created`), or the client (or plug-in) that created the statement (`gc:creator`, which is a sub-property of `dc:creator`).

Figure 2 shows the administration interface of GUMF, in particular the configuration of the dataspaces. Administrators can add plug-ins to a dataspace (cf. *"add plug-in"*) and adjust which client applications are allowed to access the dataspace (cf. *"Subscriptions"*). Via GUMF's RESTful API, client applications send advanced SPARQL queries or queries based on simple patterns. As an example, `../ds/13/predicate/interest` would return all Grapple statements in the dataspace `../ds/13` on user interests. The output format of a query can be selected as well. At the moment, GUMF supports RDF/XML, RSS 2.0 and SPARQL Query Results XML format.

In our demonstration at ABIS, we will show how GUMF is applied to mash up and reason with user profile information from different tagging and social networking systems (Flickr, Facebook, TagMe![7], and GroupMe![8]).


GUMF – User Modeling 2.0 – Dataspace
http://semweb.kbs.uni-hannover.de:8082
Google
Howdy Fabian! | home | logout
GUMF User Modeling 2.0 :)
[client applications] [dataspaces] [help]
Tag Cloud Aggregator
add
Done

Dataspace Details [details of the dataspace configuration]

id:	10	[the GUMF ID of the dataspace]
uri:	http://semweb.kbs.uni-hannover.de:8082/grapple-umf/ds/10	[the GUMF URI of the dataspace]
name:	GroupMe! – Dataspace	
description:	Default dataspace of client application GroupMe!	
admin:	Fabian	
contributor:	GroupMe!	[applications allowed to write into the dataspace]
plug-ins:	*no plug-ins plugged in yet*	[plug-ins enrich the data stored in the dataspace]
add plug-in:	Tag Cloud Aggregator add	**plug-in description:** Aggregates tag clouds from different services (e.g. the user tag cloud from Flickr GroupMe! and TagMe!) [url: http://tagme.groupme.org/tagclouds]

Subscriptions [client applications that subscribed to the this dataspace]

About subscriptions: Client applications that subscribed to the dataspace are allowed to access (read) the data made available via the dataspace.

Subscribers:

- TagMe! delete

Figure 2: Configuration of Dataspaces and Plug-Ins in the GUMF Web Application.

References

[Abel *et al.*, 2009a] F. Abel, D. Heckmann, E. Herder, J. Hidders, G.-J. Houben, D. Krause, E. Leonardi, and K. van der Slujis. A Framework for Flexible User Profile Mashups. In Proc. of *Int. Workshop on Adaptation and Personalization for Web 2.0 at UMAP '09*, Trento, Italy, 2009.

[Abel *et al.*, 2009b] F. Abel, D. Heckmann, E. Herder, J. Hidders, G.-J. Houben, D. Krause, E. Leonardi, and K. van der Slujis. Definition of an appropriate User profile format. Technical Report D2.1, Grapple project, March 2009.

[Abel *et al.*, 2008] F. Abel, N. Henze, D. Krause, and D. Plappert. User modeling and user profile exchange for Semantic Web applications. In Proc. of *Workshop on Adaptivity and User Modeling in Interactive Systems*, Wuerzburg, Germany, 2008.

[Frias-Martinez *et al.*, 2005] E. Frias-Martinez, G. Magoulas, S. Chen, and R. Macredie. Modeling human behavior in user-adaptive systems: Recent advances using soft computing techniques. *Expert Systems with Applications* **29**:320–229, 2005.

[Heckmann *et al.*, 2005] D. Heckmann, T. Schwartz, B. Brandherm, M. Schmitz, and M. von Wilamowitz-Moellendorff. GUMO - The General User Model Ontology. In Proc. of *Int. Conf. on User Modeling*, Edinburgh, UK, 428–432, 2005.

[Kobsa *et al.*, 2001] A. Kobsa, J. Koenemann, and W. Pohl. Personalized hypermedia presentation techniques for improving customer relationships. *The Knowledge Engineering Review* **16** **(2)**:111–155, 2001.

[Korth and Plumbaum, 2007] A. Korth and T. Plumbaum. A framework for ubiquitous user modeling. In Proc. of *Int. Conf. on Information Reuse and Integration*, 2007.

[4] http://www4.wiwiss.fu-berlin.de/bizer/silk/

[5] http://xmlns.com/foaf/spec/

[6] http://web-semantics.org/ns/opensocial/

[7] http://tagme.groupme.org

[8] http://groupme.org

Recommend me a Service: Personalized Semantic Web Service Matchmaking

Anna Averbakh, Daniel Krause, Dimitrios Skoutas
L3S Research Center
Hannover, Germany
{averbakh,krause,skoutas}@l3s.de

Abstract

In the Semantic Web the discovery of appropriate Semantic Web Services for a given service request, the so-called matchmaking, is a crucial task in order to bring together Web Service provider and users in an automatic manner. While most of the current matchmaking algorithms focus on purely syntactic or semantic similarity or a combination of both (hybrid approaches), the user is not taken into account in the matchmaking process itself. Hence, specific preferences and needs of a user are not taken into account in the matchmaking process.

In this paper we show how users can be engaged in the matchmaking process by providing Web 2.0 interaction to collect user feedback. Furthermore, we present the ongoing work of the integration of collaborative filtering algorithms into the matchmaking process to generate personalized matchmaking results.

1 Introduction

In today's Web, two emerging trends can be observed: On the one hand, a growing number of Web sites adheres to Web 2.0 principles [O'Reilly, 2005] and allows users generate, share, tag, comment or rate content. Hence, applications can structure and personalize large content repositories by utilizing the power of the masses. On the other hand, more and more machine-readable data in RDF format, enriched by meta-data, is available on the Web – Sindice[1], for example, has indexed more than 50 million Web pages containing RDF data – fostering machine-machine interaction on the Web.

Both trends, as stated by Tim Berners-Lee [Berners-Lee, 2006], fit well together: Web 2.0 applications, offering simpler and interactive user interfaces, can be combined with the intelligence of the Semantic Web. A key task to bring both, Web 2.0 and Web 3.0 together is the matchmaking of Semantic Web Services: User interfaces rely on the functionality offered by Web Services and hence need to find high quality Web Services, while the matchmaking process for such Web Services can highly benefit from the interaction with the user.

The matchmaking process itself fulfills the main requirements to apply personalization: a) the available amount of Web Services is too big to be browseable and b) the requirements of users are heterogeneous. However, most of today's state-of-the-art algorithms do not take the user into account.

In this paper, we present an generic approach, that utilizes various semantic and syntactic matchmakers to apply collaborative filtering algorithms to the matchmaking process of Web Services.

2 Scenario

Alice, Paul and Bob love music and enjoy listening to music in their favorite Web 2.0 application. All of them have different favors for the kind of music: While Alice likes classics best, Paul enjoys jazz most and Bob is addicted to rock music. In order to select songs that fit best to the favors of the listeners, the Web 2.0 application invokes Web Services to receive music recommendations.

In our scenario, three different recommender services are available, that recommend music. Two of them - due to a limited datasource - can only recommend specific kinds of music, jazz music and rock music, while the third service is a general purpose music recommender service.

A non-personalized application would specify a request that matches generic music recommender services and hence would not receive the jazz and rock music recommender as first matchmaking result – even though we consider them as optimal for Paul and Bob. To receive personalized matchmaking results, an application needs to adapt the service requests according to a user's preferences. Thus, applications need to have a quite precise knowledge of the domain in order to specify appropriate requests. Furthermore, applications have to rely on properly specified and interlinked ontologies: For example, a service request for classical music recommenders can only be answered in our scenario if classical music is explicitly specified as subconcept of music – elsewise the general purpose music service could not be identified as a match.

As enabling personalized matchmaking by adapting the service request is not an optimal solution becomes more obvious if a fourth service occurs in the system that also recommends rock music but delivers much better recommendations. Both rock music recommender services have the same service description and hence cannot be distinguished by current matchmakers.

However, Alice, Paul and Bob enjoy Web 2.0 features and comment, bookmark and rate the music that they have listened to. In order to utilize these additional information about a user, a matchmaking architecture is needed that stores user feedback and utilizes it in the matchmaking process.

[1] http://www.sindice.com/

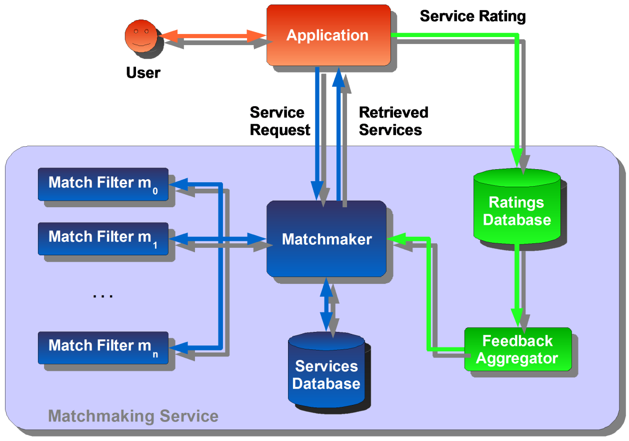


Figure 1: Personalized matchmaking service with feedback component

3 Architecture

Our system was build as a service oriented architecture with the matchmaking service (see Figure 1) as central component. This matchmaking service can by invoked by applications to receive a personalized list of matching services to a given service request. Therefore, the service invokes match filters, which implement a specific matchmaking algorithm, e.g. extended Jaccard similarity coefficient, cosine similarity etc. (see [Klusch *et al.*, 2006]) and a feedback aggregator that assigns a score to the services based on the previously given rating of the users.

After the matchmaking service returned a list of services, users can assign new rating, expressing whether the service was appropriate for the given service request or not. As the task of rating the quality of a service in respect to a given request is not trivial for a user, applications can assist the user in order to give precise feedback. For example, if an application uses services to generate music recommendations, then users can be asked whether they consider the given recommendations appropriate. Based on the assumption that services delivering high quality recommendations are better matches for this task, the application can infer the relevance of a service, and pass this information as a user rating to the matchmaking service. Such user ratings are stored in the ratings database and can be accessed by the feedback aggregator.

More detailed information about the architecture can be found at [Averbakh *et al.*, 2009].

4 Personalized Matchmaking

For personalized matchmaking, we use a domination-based matchmaking approach, as described in [Skoutas *et al.*, 2009]. This approach uses the skyline algorithm [Kossmann *et al.*, 2002] to combine multiple matchmaking metrics. Besides the existing matchmaker metrics $M_0 - M_4$ from the OWLS-MX matchmaker [Klusch *et al.*, 2006], we define an additional metric rec_x, that expresses whether a service shall be recommended to a user or not.

Assume that the collected user ratings are stored as a set $\mathcal{T} \subseteq \mathcal{U} \times \mathcal{R} \times \mathcal{S} \times F$ in the ratings database, where $\mathcal{U}$ is the set of all users that have provided a rating, $\mathcal{R}$ is the set of all previous service requests stored in the system, $\mathcal{S}$ is the set of all the available Semantic Web Service descriptions in the repository, and $F \in [0,1]$ denotes the user rating, i.e., how relevant a particular service was considered with respect to a given request (with higher values representing higher relevance). Thus, a tuple $T = (U, R, S, f) \in \mathcal{T}$ denotes that a user U considers the service $S \in \mathcal{S}$ to be relevant for the request $R \in \mathcal{R}$ with a score f.

The recommendation score rec_1 of a service s_1 and a given request r_1 for a specific user u_1 can be calculated as the average of the previous ratings from the user u_1 for service s_1 in respect to request r_1:

$$rec_1(u_1, s_1, r_1) = \frac{\sum_{(u_1,s_1,r_1,f)\in T} f}{|\{(u_1, s_1, r_1, f) \in T\}|} \qquad (1)$$

However, if a user specifies a request for the first time this formula is not applicable. We can overcome this new-request problem by assuming that for similar requests a user will rate services similarly.

If $SIM_r \subseteq R$ denotes a set of services requests that are considered as similar to a given service request r and $sim(r_1, r_2) \in [0,1]$ denotes the similarity value between r_1 and r_2, rec_2 is calculated by:

$$rec_2(u_1, s_1, r_1) = \frac{\sum_{x\in X} f * sim(r_1, r_2)}{|X|} \qquad (2)$$

with

$$X := \{(u_1, s_1, r_2, f) \in T : r_2 \in SIM_{r1}\} \qquad (3)$$

Hence, the more similar a request r_2 is to a given request r_1, the more important is the given feedback of s_1 to r_2 for r_1.

As the amount of available Web Services grows rapidly (already today the latest OWLS test collection[2] contains more than 1000 Semantic Web Services) the user ratings - service matrix will become very sparse. Hence, the above formula will not be applicable in many cases.

To overcome the sparsity problem, we now consider also ratings from other users SIM_u, which are considered similar to the given user u_1. We consider users to be similar if they have rated services similarly. Assume that the users are represented by their rating vector, $sim(u_1, u_2)$ denotes the cosine similarity between the two rating vectors of the users u_1 and u_2. Then, the collaborative filtering approach as presented in [Shardanand and Maes, 1995] can be applied to rec_3 by:

$$rec_3(u_1, s_1, r_1) = \frac{\sum_{y\in Y} f * sim(u_1, u_2) * sim(r_1, r_2)}{|Y|} \qquad (4)$$

with

$$Y := \{(u_2, s_1, r_2, f) \in T : r_2 \in SIM_{r1}, u_2 \in SIM_{u1}\} \qquad (5)$$

Hence, ratings from very similar users that rated a service s_1 in the context of a given request r_2 that is very similar to the request r_1 is considered as highly relevant for the recommendation score of s_1 in respect to r_1.

5 Related Work

With the advent of Semantic Web Services like WSDL-S [Akkiraju and et. al., 2005], OWL-S [Burstein and et. al., 2004] or WSMO [H. Lausen, A. Polleres, and D. Roman (eds.), 2005], the shortcomings of keyword based search offered by seekda[3] or syntactic matchmaking performed on

[2] available at http://www.semwebcentral.org/projects/owls-tc/

[3] http://seekda.com/

WSDL files war replaced by semantic matchmaking.

Based on such Semantic Web Services, some approaches already exist about involving the user in the process of service discovery. Ontologies and user profiles are employed in [Balke and Wagner, 2003], which then uses techniques like query expansion or relaxation to better satisfy user requests. However, such an approach will not solve the task to identify which of two given services with identical service descriptions to choose. The work presented in [Xu *et al.*, 2007] focuses on QoS-based Web Service discovery, proposing a reputation-enhanced model. A reputation manager assigns reputation scores to the services based on user feedback regarding their performance. Then, a discovery agent uses the reputation scores for service matching, ranking and selection. The application of user preferences, expressed in the form of soft constraints, to Web Service selection is considered in [Kießling and Hafenrichter, 2002], focusing on the optimization of preference queries. The approach in [Lamparter *et al.*, 2007] uses utility functions to model service configurations and associated user preferences for optimal service selection. In [Dong *et al.*, 2004], different types of similarity for service parameters are combined using a linear function with manually assigned weights. Learning the weights from user feedback is proposed, but it is left as an open issue for future work. Collaborative filtering for discovering Web Service registries was presented in [Sellami *et al.*, 2009]. In [Manikrao and Prabhakar, 2005] collaborative filtering is used to re-rank the matching candidates from non-personalized matchmaking algorithms. Due to the separation of matchmaking and ranking, this approach tends to prefer weak matching popular services to well-matched services.

6 Conclusion and Future Work

In this paper we outlined by a scenario that a non-personalized one-fits-all matchmaking approach will not provide optimal service results. We introduced our architecture that enables users to influence the matchmaking process by giving feedback by well-known Web 2.0 techniques. Based on previous work, we combined different semantic and syntactic matchmaking algorithms and collaborative filtering based on user feedback by a skyline approach.

As future work, we will conduct a user study in order to evaluate the performance of our personalized matchmaking approach.

References

[Akkiraju and et. al., 2005] Rama Akkiraju and et. al. Web Service Semantics - WSDL-S. In *W3C Member Submission*, November 2005.

[Averbakh *et al.*, 2009] Anna Averbakh, Daniel Krause, and Dimitrios Skoutas. Exploiting user feedback to improve semantic web service discovery. Technical report, 2009.

[Balke and Wagner, 2003] Wolf-Tilo Balke and Matthias Wagner. Cooperative Discovery for User-Centered Web Service Provisioning. In *ICWS*, pages 191–197, 2003.

[Berners-Lee, 2006] Tim Berners-Lee. Semantic web and web 2.0. conference presentation, November 2006.

[Burstein and et. al., 2004] Mark Burstein and et. al. OWL-S: Semantic Markup for Web Services. In *W3C Member Submission*, November 2004.

[Dong *et al.*, 2004] Xin Dong, Alon Y. Halevy, Jayant Madhavan, Ema Nemes, and Jun Zhang. Similarity Search for Web Services. In *VLDB*, pages 372–383, 2004.

[H. Lausen, A. Polleres, and D. Roman (eds.), 2005] H. Lausen, A. Polleres, and D. Roman (eds.). Web Service Modeling Ontology (WSMO). In *W3C Member Submission*, June 2005.

[Kießling and Hafenrichter, 2002] Werner Kießling and Bernd Hafenrichter. Optimizing Preference Queries for Personalized Web Services. In *Communications, Internet, and Information Technology*, pages 461–466, 2002.

[Klusch *et al.*, 2006] Matthias Klusch, Benedikt Fries, and Katia P. Sycara. Automated Semantic Web service discovery with OWLS-MX. In *AAMAS*, pages 915–922, 2006.

[Kossmann *et al.*, 2002] Donald Kossmann, Frank Ramsak, and Steffen Rost. Shooting stars in the sky: an online algorithm for skyline queries. In *VLDB '02: Proceedings of the 28th international conference on Very Large Data Bases*, pages 275–286. VLDB Endowment, 2002.

[Lamparter *et al.*, 2007] Steffen Lamparter, Anupriya Ankolekar, Rudi Studer, and Stephan Grimm. Preference-based selection of highly configurable web services. In *WWW*, pages 1013–1022, 2007.

[Manikrao and Prabhakar, 2005] Umardand Shripad Manikrao and T. V. Prabhakar. Dynamic selection of web services with recommendation system. In *NWESP '05: Proceedings of the International Conference on Next Generation Web Services Practices*, page 117, Washington, DC, USA, 2005. IEEE Computer Society.

[O'Reilly, 2005] Tim O'Reilly. O'Reilly Network: What is Web 2.0, September 2005.

[Sellami *et al.*, 2009] Mohamed Sellami, Samir Tata, Zakaria Maamar, and Bruno Defude. A recommender system for web services discovery in a distributed registry environment. *Internet and Web Applications and Services, International Conference on*, 0:418–423, 2009.

[Shardanand and Maes, 1995] Upendra Shardanand and Pattie Maes. Social information filtering: algorithms for automating "word of mouth". In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[Skoutas *et al.*, 2009] Dimitrios Skoutas, Dimitris Sacharidis, Alkis Simitsis, Verena Kantere, and Timos Sellis. Top-k Dominant Web Services under Multi-criteria Matching. In *EDBT*, pages 898–909, 2009.

[Xu *et al.*, 2007] Ziqiang Xu, Patrick Martin, Wendy Powley, and Farhana Zulkernine. Reputation-Enhanced QoS-based Web Services Discovery. In *ICWS*, pages 249–256, 2007.

Challenges in Developing User-Adaptive Intelligent User Interfaces

Melanie Hartmann
Telecooperation Group
Technische Universität Darmstadt
64289 Darmstadt, Germany
melanie@tk.informatik.tu-darmstadt.de

Abstract

As user interfaces become more and more complex and feature laden, usability tends to decrease. One possibility to counter this effect are intelligent user interfaces (IUIs) that support the user's interactions. In this paper, we give an overview of design challenges identified in literature that have to be faced when developing user-adaptive IUIs and possible solutions. Thereby, we place special emphasis on design principles for successful adaptivity.

1 Introduction

The increasing complexity in today's applications, e.g. the number of available options, often leads to a decreased usability of the user interface. This effect can be countered with intelligent user interfaces (IUIs) that support the user in performing her tasks by facilitating the interaction as much as possible. IUIs facilitate information retrieval by suggesting relevant information or they support the system use, e.g. by providing explanations, performing tasks for the user, or adapting the interface. In this paper, we focus on user-adaptive IUIs which are able to adapt their behavior to individual users. In our opinion this is a key feature for IUIs as the support that should be provided by an IUI heavily depends on the needs and preferences of each user.

We present here the results of a literature survey of the main design challenges that have to be faced when designing a user-adaptive IUI and list existing approaches how to cope with these challenges. We thereby focus on issues relevant for human computer interaction and not on the underlying algorithms.

The remainder of this paper is structured as follows. In Section 2, we provide a definition of user-adaptive IUIs. In Section 3, we then describe the challenges which have to be faced in developing IUIs. As adaptivity plays a crucial role for user-adaptive IUIs, we place special emphasis on the adaptivity of IUIs, and review the main results gathered from user studies that were performed for evaluating which factors influence the value of an adaptation for a user (Section 4).

2 User-Adaptive Intelligent User Interfaces

The area of IUIs is one of the most heterogeneous research subjects, covering all kinds of different disciplines, which makes it difficult to give a common definition. IUIs try to solve the standard user interface question how the interaction between user and computer can be facilitated by means of artificial intelligence. In contrast to traditional human computer interaction (HCI), IUIs do not only focus on enabling the user to perform intelligent actions but on ways to incorporate knowledge to be able to assist the user in performing actions. In contrast to traditional research in artificial intelligence (AI), IUIs do not focus on making the computer smart by itself but to make the interaction between computer and human smarter.

The goal of IUIs is to make the interaction itself as well as the presentation of information more effective and efficient to better support the user's current needs. The way to achieve this ranges from supporting a more natural interaction, e.g. by allowing multimodal or natural language input, to intelligent tutoring systems and recommender systems. Based on the definition by Maybury and Wahlster [1998], we define IUIs as follows:

> ***Intelligent User Interfaces*** *are human-machine interfaces that aim to improve the efficiency, effectiveness and naturalness of human machine interaction by representing, reasoning and acting on models of the user, domain, task, discourse, context, and device.*

In this paper, we focus on user-adaptive IUIs which are a subset of IUIs. User-adaptive IUIs hold a model for each individual user to be able to adapt its behavior accordingly, which is not necessarily the case for all IUIs as often a generic user model suffices.

3 Challenges in Developing IUIs

The main goal in developing IUIs is that they should be usable, useful and trustable [Myers, 2007]. This aligns with the main challenges identified by Maes [1994]: *Presentation*, *Competence* and *Trust*. **Presentation** is concerned with the human computer interaction part of IUIs, whereas **Competence** focuses on the artificial intelligence techniques that can be applied. The development of IUIs has to take special care of **Trust**, as the user is not willing to delegate tasks to an IUI she does not trust, thus rendering the IUI useless. However, for IUIs it is much more challenging to induce user's trust in the system than for traditional user interfaces, because IUIs apply artificial intelligence techniques whose results can often not be directly foreseen by the user and thus reduce the user's feeling of being in control of the system. In the following, we point out for each of these issues which challenges have to be faced when developing user-adaptive IUIs and describe possible ways to cope with them. An overview of the identified challenges is given in Table 1. The challenges are not disjunctive and heavily interrelated, they should just give some of the focus points for developing user-adaptive IUIs. Further, this list is not meant to be complete and not all challenges have to be faced in each IUI, e.g. collaborative filtering systems usually do not have to cope with the problem of few usage data.

Presentation	Interaction design
	Unobtrusiveness
	Adaptivity
Competence	Few usage data
	Changing user behavior
	Accuracy
Trust	Controllable behavior
	Intelligibility
	Privacy

Table 1: Challenges in developing user-adaptive IUIs

Presentation

For the presentation of IUIs, we at first need to consider how to **design the interaction** between the user and the IUI. Many IUIs are also augmentations of existing user interfaces, thus they have to be integrated into the existing layout offering the user a way to communicate with the IUI itself. Thereby, the IUI should not hamper the normal usage of the application. The interaction should also support some kind of forgiveness that is allowing the user to easily correct previously performed actions using an undo capability [Apple, 2008]. Further, the design of the interaction tackles whether and how the user can instruct the IUI [Norman, 1994] or whether an anthropomorphic agent is used for allowing the user to communicate with the IUI [Wexelblat and Maes, 1997]. These issues are closely related to trust issues that will be discussed later, i.e. how the user can control the system and which expectations are raised by the IUI.

Another important factor regarding the presentation is **unobtrusiveness** [Jameson, 2007; Langley and Fehling, 1996]. The intelligent support should not distract the user from normal usage of the application. A counterexample for this factor is Microsoft's Office Assistant that is constantly moving and thus drawing the user's attention to it without providing any relevant help for the user's current task. Wexelblatt and Maes [1997] propose to reduce the distraction of the user by minimizing the amount of interruptions and deferring interruptions until they are less disruptive. Another way to cope with this issue is to support different levels of obtrusiveness (or proactivity) depending on the information importance or the certainty in the action (e.g. applied by [Maes, 1994; Horvitz, 1999; Hartmann *et al.*, 2009]).

Furthermore, the user-adaptive IUI should be able to **adapt** its presentation to different users, devices and situations. For example, a novice user needs more explanations than an expert user and voice output is perhaps suitable for mobile usage, but not if she is sitting in a library. Further, as the interaction costs for interacting with applications via a mobile phone are much higher than in a traditional desktop setting, more support may be desirable in these settings. However, adaptivity does not only influence the presentation of an IUI, but also how much the user trusts the system or which demands it puts on the underlying algorithms (i.e. affecting the competence of the IUI). We review the main findings from studies regarding which factors influence the value of an adaptation in Section 4.

Competence

The competence of an IUI is determined by the underlying algorithms. However, as we focus in this paper on human computer interaction issues, we only provide here a short overview of the main challenges that have to be faced for the competence of a system.

At first, many user-adaptive IUIs cannot rely on a huge amount of training data at the beginning, especially if they need training data for each individual user. Thus, the algorithms used for user-adaptive IUIs mostly have to be able to deal with **few usage data**. For that reason, systems that just learn from observation are usually not of great aid at the beginning ("slow-start problem"). This problem can be faced e.g. by relying on predefined models or by using a default model that is inferred from the models of other users. However, the former requires great modeling effort by a developer and the latter can cause privacy problems (as discussed below).

A second problem that arises is that the **user's behavior changes** over time [Höök, 2000]. Especially when she starts interacting with an application as novice, her usage patterns as expert will later dramatically differ from the initial patterns. For that purpose, ageing can be used that weighs older interactions as less important than more recent interactions.

Finally, in order to be beneficial for the user, the artificial intelligence of course needs to produce correct results with a high **accuracy**. This is especially important as erroneous support can easily lead to losing the user's trust [Leetiernan *et al.*, 2001].

Trust

The trust the user puts in an IUI is influenced by many factors especially by presentation issues as discussed before. In the following, we state the main challenges identified in literature that have to be considered when building trustable IUIs. At first, it is essential that the user feels in **control** of the system. The user should be able to correct and adjust the IUI's actions and to control its autonomy [Höök, 2000; Bellotti and Edwards, 2001; Glass *et al.*, 2008; Dey and Newberger, 2009]. One possibility to control the system's actions is to require the user to approve or disapprove the system's actions [Cypher, 1991] or by letting the user specify confidence thresholds for actions [Maes, 1994]. However, giving the user the maximal control at all times is usually not desirable as the users differ in their desire for control [Jameson and Schwarzkopf, 2002] and too much control may lead to distraction and time-wasting [Kay, 2001]. The amount of control should also depend on the criticalness of the task, e.g. for non-critical tasks, like prefilling data in input fields, a lower level of control is needed than for automatically buying goods. Another factor that influences how much control the user wants to exert is her trust in the system that (hopefully) evolves over time. For all those reasons, an IUI should support variable levels of control that can also be adjusted by the user.

Another important issue for establishing the user's trust in the system is **intelligibility**, i.e. to enable the user to understand the system's actions [Bellotti and Edwards, 2001; Dey and Newberger, 2009]. As stated by Maes [1994] a user more likely trusts an IUI if she sees in advance what the agent would do. One way to achieve intelligibility is *transparency*, i.e. that the IUI helps the user understand its actions. Transparency can be realized by an IUI for example by giving feedback of its actions [Maes, 1994], by being able to justify its actions, or by making the user aware of automatic adaptations [Hartmann *et al.*, 2009]. Another way of increasing the system's intelligibility is to give the user *access to the knowledge source* that was used for providing support [Glass *et al.*, 2008]. For example, Cook and Kay [1994] argue that the system should let the user inspect and modify the system's user models. The intelligibility of the system's actions should thus support the user to develop an appropriate model of the IUI's behavior. Thereby, it is not necessary to mediate a complete model of the IUI, as "understanding comes from a careful blend of hiding and revealing [system] state and functioning" [Wexelblat and Maes, 1997]. They argue that for example for driving a car, it is also not necessary to have a complete model of how the engine or the breaks work. This blend can be achieved by applying a black box in a glass box system [Höök *et al.*, 1996], i.e. complex inferences are hidden from view in a black box system, whereas a simpler model is conveyed to the user, e.g. cartoons illustrating the system's state as used by [Kozierok and Maes, 1993]. Another factor influencing the intelligibility of the system is how *predictable* the system's actions are perceived by the user and finally which *expectations* she poses in the system [Glass *et al.*, 2008]. Erroneous higher expectations can easily lead to disappointing the user and thus stopping the user from using the system. This is also one of the main reasons why many researchers argue against using anthropomorphic agents for communicating with IUIs (e.g. [Shneiderman, 1997]), as they are perceived by the user to be similar to a human being and that they thus could also take responsibility for their actions.

Finally, for IUIs that share information between users, **privacy** has to be regarded. Thereby the requirements that are posed on privacy differ between applications. For example, the users of FireFly[1], an application for sharing preferences for music or movies, did not perceive this sharing as critical, whereas users of the Doppelgänger system [Orwant, 1994] that provides personalized news which also considered the news that a colleague is usually reading, had strong privacy concerns against the system. This might be the case as the data differed in their level of importance to the user and as the data was not anonymized in the Doppelgänger system in contrast to the Firefly system. Besides anonymizing the data, another solution proposed for this problem is to split the user model in a private and a public part [Cook and Kay, 1994].

4 Adaptivity Challenges

There has been a debate for years whether automatic adaptation optimizes the user interface or disorients the user [Greenberg and Witten, 1985; Mitchell and Shneiderman, 1989; Shneiderman and Maes, 1997]. The IUI community often favors automatic adaptivity, whereas the HCI community tends towards adaptable approaches that allow the user to customize her interface herself without any automatism. However, many studies have shown that users often fail to use the offered adaptation mechanisms [Oppermann and Simm, 1994], and when they do, they often do not re-customize it if their working habits change [McGrenere *et al.*, 2002].

The value of an adaptation for a user is usually measured as the user interface's usability, i.e. the user's efficiency, effectiveness and her satisfaction. Findlater and McGrenere [2008a] propose to take another factor into account when evaluating adaptive user interfaces: the user's awareness of advanced features. They noted that increased efficiency can lead to a decreased awareness of advanced features, probably because the adaptivity allows them to focus more on the task itself and not on the available menu elements [Findlater and McGrenere, 2008b]. Thus it can hamper the learning of novel user interfaces. However, for seldom used applications or applications in which the user is already an expert, no awareness of advanced features is needed.

In this section, we focus on the usability aspect and summarize the main results from user studies reported in literature. They investigate when an adaptation is useful and how adaptation has to be designed to improve the usability and to avoid confusion. The identified factors thereby comprise presentation as well as competence issues and also influence the trust in the system. An overview of these factors can be found in Table 2.

Presentation	Spatial Stability
	Locality
Competence	Accuracy
	Predictability
Further factors	Interaction frequency
	Task Complexity
	Average interaction costs

Table 2: Factors influencing the value of an adaptation for a user

Regarding the presentation of the adaptation, Gajos *et al.* [2006] found that **spatial stability** increases the user satisfaction and that **high locality** improves discoverability of the adaptation, i.e. that the promoted user interface element appears close to its original position. The spatial stability is required to enable the user to maintain a mental model of the application. An example for spatial in/stability are the Smart Menus used in Microsoft Office, in former versions they hid infrequently used items from view, which caused many negative reactions among users due to their spatial instability, whereas in Office 2007 the menus contain predefined adaptive parts (e.g. displaying the most recently used items) which seems to have much more supporters.

Another important factor is the behavior of the algorithm that adapts the UI, i.e. its **accuracy** and its **predictability** (which is of course closely related to the accuracy issue discussed before). Gajos *et al.* [2008] found that an increase in each of the factors leads to a strongly improved user satisfaction. An increased accuracy moreover leads to improved user performance and more frequent use of the adaptive part. The increase in accuracy thereby had stronger effects on user performance, on how often they utilized it, and on some satisfaction ratings. Another study by Tsandilas and Schraefel [2005] also showed that participants performed faster and utilized the adaptive parts more often at higher accuracy levels. Further, they noted that users tend to underestimate the accuracy of algorithms when the algorithm has a low accuracy (in the study many users estimated a 60% accuracy with an accuracy of under 50%). Findlater and McGrenere [2008b] also showed that the accuracy influences the user's perception of the algorithm's predictability. Higher accuracy user interfaces were perceived as more predictable and consistent.

Gajos *et al.* [2006] state that the **interaction frequency**

[1] A company founded by a group of engineers from MIT media lab including Pattie Maes and sold to Microsoft in April 1998.

and the **task complexity** also play a role in the perceived value of the adaptation. If the task is rather simple and largely mechanical interactions need to be performed, the locality of the adaptation plays a more important role than for more complex tasks. Further, users are more likely able to build mental models for applications which are of low complexity or with which they frequently interact. These mental models can reduce the positive effect of adaptive parts if the interaction costs for using the unadapted version and the adapted version do not differ much, e.g. the amount of required clicks is about the same.

Another factor that is noted by Gajos *et al.* [2006] is the frequency of the adaptation. However, there was no study that directly compared the influence of adaptation frequencies; this was just concluded from two studies about split menus[2] by Sears and Shneiderman [1994] and Findlater and McGrenere [2004] that differed in the adaptation frequency and also in the received results. They both compare non-adaptive menus to an adaptive split menu, whereby Sears and Shneiderman adapt the elements that are displayed only once per user and session, whereas the interface by Findlater and McGrenere can adapt up to once per interaction. Sears and Shneiderman found that the users were faster and more satisfied with the adaptive version, in contrast to the findings by Findlater and McGrenere. However, the static version that was used by Findlater and McGrenere is also a split menu containing the most relevant commands in the usually adaptive part. Thus, it is similar to the adaptive menu used by Sears and Shneiderman and is already the best possible single menu for the experimental task. Thus, the adaptive menu in the experiment by Findlater and McGrenere did not have much of a chance (see also [Jameson, 2007]). For that reason, we state that the adaptation frequency just influences the spatial stability and the predictability of an algorithm, but is no factor on its own.

A final factor that was not considered by Gajos *et al.* [2006] are the average **interaction costs**. This factor is especially important in the area of ubiquitous computing where the interaction costs can heavily vary as not only standard desktop computers with large screen, mouse and keyboard are used, but also small screen devices like mobile phones and various input devices like a Wii. Findlater and McGrenere [2008b] recently showed in a user study that interaction on small screen devices benefits more from adaptive menus than interaction on large screens. They showed that the user's performance and the utilization of the adaptive parts increased more for small than for large screen devices compared to their static counterparts.

5 Summary

In this paper, we gave an overview of the major design challenges that have to be faced when developing user-adaptive IUIs. In summary, the following challenges were identified: Regarding the presentation of intelligent support mechanisms, special attention has to be paid to the design of the interaction, that it does not disrupt the user's normal workflow and that it is adapted to the specific user needs and the given situation. The algorithms for computing the support have to be able to cope with few usage data and changing user behavior. They should be able to provide accurate support whenever possible as erroneous support has a strong adverse effect on the user's trust in the system. Other factors that influence the user's trust are whether the user feels in control of the system, whether she can understand the system's behavior and whether the user's privacy is sufficiently protected.

As adaptivity plays an important role in user-adaptive IUIs, we also reviewed the major factors that influence the benefit of a user interface adaptation. At first, the adaptivity should be realized in predefined areas, so that the greater part of the user interface remains stable. If the adaptation relocates user interface elements, this should be realized close to their original position to facilitate the discoverability. Further, the algorithm used for the adaptation should provide highly accurate results and the results should be predictable for the user. If the user knows how to interact with an application (i.e. interaction frequency and the task complexity), this can lower the additional benefit which can be yield from an adaptation. Finally, the interaction costs influence the benefit of an adaptation: the higher the interaction costs, the more does an adaptation pay off.

References

[Apple, 2008] Apple. Apple human interface guidelines, 2008.

[Bellotti and Edwards, 2001] Victoria Bellotti and Keith Edwards. Intelligibility and accountability: human considerations in context-aware systems. *Human-Computer Interaction*, 16(2):193–212, 2001.

[Cook and Kay, 1994] R. Cook and J. Kay. The justified user model: a viewable, explained user model. In *4-th International Conference on User Modeling, Hyannis, MA*, pages 145–150, 1994.

[Cypher, 1991] Allen Cypher. EAGER: programming repetitive tasks by example. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, pages 33–39, New Orleans, Louisiana, United States, 1991. ACM.

[Dey and Newberger, 2009] Anind K. Dey and Alan Newberger. Support for context-aware intelligibility and control. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 859–868, Boston, MA, USA, 2009. ACM.

[Findlater and McGrenere, 2004] Leah Findlater and Joanna McGrenere. A comparison of static, adaptive, and adaptable menus. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 89–96, Vienna, Austria, 2004. ACM.

[Findlater and McGrenere, 2008a] Leah Findlater and Joanna McGrenere. Comprehensive user evaluation of adaptive graphical user interfaces. In *Usable AI*, 2008.

[Findlater and McGrenere, 2008b] Leah Findlater and Joanna McGrenere. Impact of screen size on performance, awareness, and user satisfaction with adaptive graphical user interfaces. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1247–1256, Florence, Italy, 2008. ACM.

[Gajos *et al.*, 2006] Krzysztof Z. Gajos, Mary Czerwinski, Desney S. Tan, and Daniel S. Weld. Exploring the design space for adaptive graphical user interfaces. In *Proceedings of the working conference on Advanced visual interfaces*, pages 201–208, Venezia, Italy, 2006. ACM.

[2] A split menu is a menu that contains an adaptive part that is clearly separated from the rest of the menu (e.g. used for selecting the font in Microsoft Office 2007)

[Gajos *et al.*, 2008] Krzysztof Z. Gajos, Katherine Everitt, Desney S. Tan, Mary Czerwinski, and Daniel S. Weld. Predictability and accuracy in adaptive user interfaces. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1271–1274, Florence, Italy, 2008. ACM.

[Glass *et al.*, 2008] Alyssa Glass, Deborah L. McGuinness, and Michael Wolverton. Toward establishing trust in adaptive agents. In *Proceedings of the 13th international conference on Intelligent user interfaces*, pages 227–236, Gran Canaria, Spain, 2008. ACM.

[Greenberg and Witten, 1985] Saul Greenberg and Ian H. Witten. Adaptive personalized interfaces—A question of viability. *Behaviour & Information Technology*, 4(1):31, 1985.

[Hartmann *et al.*, 2009] Melanie Hartmann, Daniel Schreiber, and Max Mühlhäuser. Providing Context-Aware interaction support. In *Proceedings of Engineering Interactive Computing Systems (EICS)*, page to appear. ACM, 2009.

[Horvitz, 1999] Eric Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, pages 159–166, Pittsburgh, Pennsylvania, United States, 1999. ACM.

[Höök *et al.*, 1996] Kristina Höök, Jussi Karlgren, Annika Wærn, Nils Dahlbäck, Carl Jansson, Klas Karlgren, and Benoît Lemaire. A glass box approach to adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2):157–184, July 1996.

[Höök, 2000] K. Höök. Steps to take before intelligent user interfaces become real. *Journal of Interacting with Computers*, 12(4):409?426, February 2000.

[Jameson and Schwarzkopf, 2002] Anthony Jameson and Eric Schwarzkopf. Pros and cons of controllability: An empirical study. In *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 193–202. Springer-Verlag, 2002.

[Jameson, 2007] Anthony Jameson. Adaptive interfaces and agents. In Julie A. Jacko and Andrew Sears, editors, *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*, pages 305–330. Lawrence Erlbaum Associates, Inc., 2007.

[Kay, 2001] Judy Kay. Learner control. *User Modeling and User-Adapted Interaction*, 11(1-2):111–127, 2001.

[Kozierok and Maes, 1993] Robyn Kozierok and Pattie Maes. A learning interface agent for scheduling meetings. In *IUI '93: Proceedings of the 1st international conference on Intelligent user interfaces*, page 81?88, New York, NY, USA, 1993. ACM.

[Langley and Fehling, 1996] Pat Langley and Michael Fehling. *The Experimental Study of Adaptive User Interfaces*. 1996.

[Leetiernan *et al.*, 2001] Scott Leetiernan, Edward Cutrell, Mary Czerwinski, and Hunter Hoffman. Effective notification systems depend on user trust. 2001.

[Maes, 1994] Pattie Maes. Agents that reduce work and information overload. *Commun. ACM*, 37(7):30–40, 1994.

[Maybury and Wahlster, 1998] Mark T. Maybury and Wolfgang Wahlster, editors. *Readings in intelligent user interfaces*. Morgan Kaufmann Publishers Inc., 1998.

[McGrenere *et al.*, 2002] Joanna McGrenere, Ronald M. Baecker, and Kellogg S. Booth. An evaluation of a multiple interface design solution for bloated software. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, pages 164–170, Minneapolis, Minnesota, USA, 2002. ACM.

[Mitchell and Shneiderman, 1989] J. Mitchell and B. Shneiderman. Dynamic versus static menus: an exploratory comparison. *SIGCHI Bull.*, 20(4):33–37, 1989.

[Myers, 2007] Brad A. Myers. A user acceptance equation for intelligent assistants. In *AAAI 2007 Spring Symposium on Interaction Challenges for Intelligent Assistants*, 2007.

[Norman, 1994] Donald A. Norman. How might people interact with agents. *Commun. ACM*, 37(7):68–71, 1994.

[Oppermann and Simm, 1994] Reinhard Oppermann and Helmut Simm. *Adaptability: user-initiated individualization*, pages 14–66. Lawrence Erlbaum Associates, Inc., 1994.

[Orwant, 1994] Jon Orwant. Heterogeneous learning in the doppelgänger user modeling system. *User Modeling and User-Adapted Interaction*, 4(2):107–130, June 1994.

[Sears and Shneiderman, 1994] Andrew Sears and Ben Shneiderman. Split menus: effectively using selection frequency to organize menus. *ACM Trans. Comput.-Hum. Interact.*, 1(1):27–51, 1994.

[Shneiderman and Maes, 1997] Ben Shneiderman and Pattie Maes. Direct manipulation vs. interface agents. *interactions*, 4(6):42–61, 1997.

[Shneiderman, 1997] Ben Shneiderman. Direct manipulation for comprehensible, predictable and controllable user interfaces. In *IUI '97: Proceedings of the 2nd international conference on Intelligent user interfaces*, pages 33–39, New York, NY, USA, 1997. ACM.

[Tsandilas and m. c. Schraefel, 2005] Theophanis Tsandilas and m. c. Schraefel. An empirical assessment of adaptation techniques. In *CHI '05 extended abstracts on Human factors in computing systems*, pages 2009–2012, Portland, OR, USA, 2005. ACM.

[Wexelblat and Maes, 1997] Alan Wexelblat and Pattie Maes. Issues for software agent UI. *Unpublished Manuscript*, 1997.

Analyzing Client-Side Interactions to Determine Reading Behavior

David Hauger[1] and **Lex Van Velsen**[2]

[1]Institute for Information Processing and Microprocessor Technology
Johannes Kepler University, Linz, Austria

[2]Department of Technical and Professional Communication
University of Twente, Enschede, Netherlands

Abstract

Traditional monitoring and user modeling techniques in adaptive hypermedia systems consider pages as atomic units although different sections may refer to different concepts. This has been mainly due to the fact that most user interactions being monitored referred to the request of a new document and there was too little activity information to differentiate between sections of a page. Client-side monitoring can provide additional information on user interactions inside the browser window and may relate them to areas within a document. A user study was carried out to show whether and how this data might be used to identify which parts of a page have been read.

1 Introduction

It has been a widely accepted fact for several years now that "the user can prefer some nodes and links over others and some parts of a page over others" [Brusilovsky, 1996]. Opening a page does not necessarily mean that a user read all its contents. Consequently, adaptive hypermedia systems (AHS) should monitor these nodes separately to tell (a) how much of a page has been read, and (b) what parts of a page have been read or are of particular interest, especially if they concern different topics.

Most AHS try to (partially) meet these demands by monitoring requests to the server, which makes it possible to determine the links a user followed. Nevertheless, concerning text nodes (or links that have not been followed), most AHS treat pages as atomic items. Elaborate algorithms try to add additional information to user models by analyzing requests (e.g. to calculate the estimated "time spent reading" based on the time difference between requests [Farzan and Brusilovsky, 2005]), but there are hardly any attempts to treat different parts of a page separately [Hauger, 2008].

The approach put forward in this paper shows how monitoring user interactions inside the browser could help to overcome these limitations. A user study has been carried out to determine how users interact and how it is possible to determine whether a page has been read.

2 Related Work and State of the Art

Traditional user modeling techniques of AHS log requests of resources on the server and use this information as a basis for modeling. However, most interactions of users do not cause requests to the server (mouse movements, scrolling, etc.) and are therefore not monitored.

Several attempts have been made to use client-side interactions in AHS. Hijikata [Hijikata, 2004] showed that text tracing, link pointing, link clicking and text selection are an indicator for interest. Goecks and Shavlik [Goecks and Shavlik, 2000] defined a "level of activity" based on mouse and scrolling activities monitored via JavaScript. They used it for a neural network inside the browser. Hofmann et al. [Hofmann *et al.*, 2006] sent timestamps of interactions to the server to calculate periods of inactivity.

Claypool et al. [Claypool *et al.*, 2001] developed "The Curious Browser" to log interaction events inside the browser. The results were used to establish a connection between user interaction and the level of interest. Although this solution is effective, it is not ideal because in order to be able to use client-side information in common e-learning situations, additional hardware and software requirements should be avoided and standard technologies should be used for monitoring and transmitting the data. Putzinger [Putzinger, 2007] used mouse and keyboard events on input elements to determine the "locus of attention". This information has been sent to the server to adaptively provide help.

Nevertheless, most systems referred to pages as a whole. Differentiating between sections requires new monitoring techniques. Eye-tracking is one possibility to identify the locus of attention [Conati *et al.*, 2007]. As the applicability of this approach is limited due to additional hardware and software requirements, other solutions using standard technologies need to be found.

Client-side user monitoring as described in [Hauger, 2009] is able to (a) retrieve additional information on user interactions and (b) treat different sections of a page separately. The work described in this paper tries to find out whether and how the information that can be retrieved may be used to determine which parts of a page have been read.

3 Client-Side User Monitoring

In order to overcome the limitations of traditional approaches using server-side logs as the only source of information, the monitoring process itself could be improved by monitoring activities within the browser window [Hauger, 2008]. For this reason a JavaScript library has been developed which monitors these client-side events and maps them to parts of a page [Hauger, 2009].

3.1 Page Fragmentation

Different sections of a page in an AHS might need to be treated separately. As exact mouse positions might be difficult to compare and evaluate, alternative segmentation techniques need to be considered that are robust to changes in the size and topology of page elements. The library that has been developed supports different approaches to split pages:

- *split by vertical position*: Independently from the actual content a page may be vertically divided into k segments; each one representing $\frac{1}{k}$ of the page. This type of segmentation may be used to calculate how much of a document has been read and it may easily be applied for static and unstructured pages.
- *split by content type*: In oder to identify learning style preferences it is for example possible to monitor images separately to make assumptions on whether users prefer textual or graphical content.
- *split by semantic meta data available*: If there is already semantic meta data available (concepts, keywords, etc.), it is possible to monitor items including such additional information and relate the activity information to this data.
- *split by source*: For "composed" pages with items derived from multiple sources it is possible to automatically link user interactions to the original source of the fragment.
- *split by structural information*: Structural information (if available) like headlines may be used to distinguish between different sections of a page.
- *add custom fragments*: In addition to all mentioned splitting techniques, each HTML element may (even at runtime) be manually defined as a fragment that has to be monitored.

3.2 Monitored Interactions

As JavaScript is used to monitor interactions, the library logs the events already available (including mouse moves, clicks, keyboard activities, scrolling, window resizing and window events like focus and blur) and uses them for further processing (e.g. for mapping positions to fragments). In addition to those predefined JavaScript events, a number of custom events has been created; e.g. to identify text selections (which may be used to identify text tracing) and inactivity (no interactions for a longer period of time), as well as to determine events of a temporal basis. The monitored variables in detail:

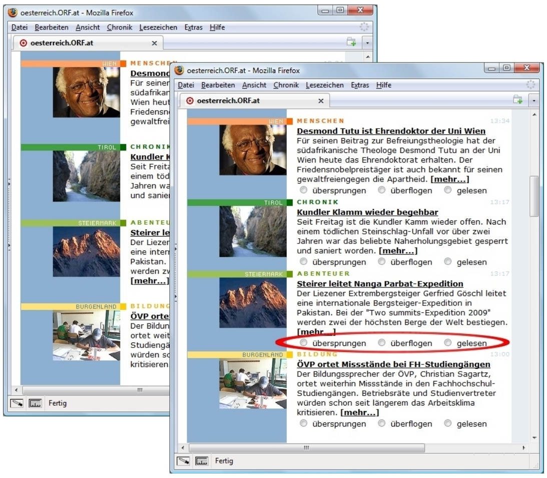
- *visible time*: The time a fragment has been visible on the screen. This can be regarded as a requirement for reading. Printing a page, saving it for offline use, etc. may allow to read parts of a text never having been visible within the browser window, but this may be regarded as an exception.
- *mouse over time*: The total time the mouse has been placed above a specified fragment. Some people place their mouse above the text they are currently reading. Therefore this is being monitored to check whether it can really be used as indicator to identify reading.
- *mouse on same y time*: The total time the mouse has been placed within the vertical borders of a fragment. This is similar to the "mouse over time", but ignoring the horizontal position of the mouse. If there is only one ("main") column of text (as in the current experiment), the two variables should be similar. For two or more columns there might be differences, e.g. if a user always places the mouse on the right side of the screen, independent from the horizontal position at which the user is reading. This, however, will be part of future work.
- *number of mouse moves*: Amount of mouse moves taking place above the current fragment. Mouse moves within 500ms have been regarded as a single mouse movement. Passing an item with the mouse in less than half a second has not been counted.
- *number of clicks*: The total amount of clicks performed on the fragment.
- *number of text selections*: Counting how often a user has selected text within a specific fragment.

The main premise of the work described in this paper is that based on these interactions it should be possible to draw additional assumptions on users' reading behavior, interests, etc.

4 User Study

In order to determine how client-side user interactions and reading behavior are related, a user study with 53 volunteers has been carried out. The results of client-side user monitoring should be compared to explicit feedback given by the users. The main goal was the identification of client-side user activities that may be used to identify which parts of a page have been read.

A single page containing a number of news items (20–23) from an Austrian news page (http://oesterreich.orf.at) has been provided. Each item consisted of a thumbnail (width: $\approx 100 - 150px$) on the left side and a headline with $4 - 6$ lines ($\approx 20 - 40$ words) of additional text (short summary of an article) next to it. Internally, the page was split automatically in order to monitor each news item separately. As the system should focus on interaction information that cannot be gained through server-side monitoring, links to the extended articles were disabled. The page was updated twice a day to increase the probability users have not read the news before, which should result in higher interest.


oesterreich.ORF.at - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
oesterreich.ORF.at
Fertig

WIEN MENSCHEN 13:34

Desmond Tutu ist Ehrendoktor der Uni Wien

Für seinen Beitrag zur Befreiungstheologie hat der südafrikanische Theologe Desmond Tutu an der Uni Wien heute das Ehrendoktorat erhalten. Der Friedensnobelpreistäger ist auch bekannt für seinen gewaltfreiengegen die Apartheid. [mehr...]

○ übersprungen ○ überflogen ○ gelesen

TIROL CHRONIK 13:17

Kundler Klamm wieder begehbar

Seit Freitag ist die Kundler Kamm wieder offen. Nach einem tödlichen Steinschlag-Unfall vor über zwei Jahren war das beliebte Naherholungsgebiet gesperrt und saniert worden. [mehr...]

○ übersprungen ○ überflogen ○ gelesen

STEIERMARK ABENTEUER 13:17

Steirer leitet Nanga Parbat-Expedition

Der Liezener Extrembergsteiger Gerfried Göschl leitet eine internationale Bergsteiger-Expedition in Pakistan. Bei der "Two summits-Expedition 2009" werden zwei der höchsten Berge der Welt bestiegen. [mehr...]

○ übersprungen ○ überflogen ○ gelesen

BURGENLAND BILDUNG 13:00

ÖVP ortet Missstände bei FH-Studiengängen

Der Bildungssprecher der ÖVP, Christian Sagartz, ortet weiterhin Missstände in den Fachhochschul-Studiengängen. Betriebsräte und Studienvertreter würden schon seit längerem das Arbeitsklima kritisieren. [mehr...]

○ übersprungen ○ überflogen ○ gelesen

Figure 1: user study: reading and evaluation page

The study itself was entirely anonymous – the participants were not even asked to enter demographic information. Participation was possible via the web. On a first page the experiment was explained and users were instructed to read only whatever interested them, as if they were visiting the news page in a normal context of use.

While they were reading, their interactions within the browser were monitored using the library mentioned in section 3. Information on the absolute location of events were mapped to the news items to be able to compare them later on. The preprocessed events were sent to the server and stored in a database, as well as the values for the variables mentioned in section 3.2 (per user and news item). In addition to this, the total time for a page being requested was recorded, which is the only information that could have been retrieved by server-side monitoring as well.

After reading the users were asked to fill a small questionnaire. For each news item they had to select whether they read this item, glanced at it or skipped it. The page for reading and the feedback form are shown in Figure 1.

It has to be stated that the feedback of the users is subjective and there may be differences in what single users regarded as reading, glancing or skipping.

The evaluation of the results should show how reading and client-side interactions within the browser are correlated. The final goal is the establishment of an algorithm that is able to tell with a satisfactory level of certainty whether a fragment has been read or not. Although the scope of the experiment is not sufficient for getting exact values and parameters for an overall algorithm, this user study should show directions towards creating it.

5 Results

A total of 53 participants completed the questions related to the news items. They provided feedback on 20 to 23 items, with an average of 22.32 items. The participants spent, on average, 2 minutes and 9 seconds reading the news page, with a standard deviation of 2 minutes and 36 seconds.

The items related to user feedback (item skipped, glanced or read) were scored in a dummy variable to enable data analysis. Each feedback option was made into a separate variable with a score of either 0 (item not skipped, not glanced or not read) or 1 (item skipped, glanced or read). The responses were based on the participants' subjective assessment of their own behavior, and thus there might be differences in what users regarded as read, skipped or glanced at. For some users, skipping meant not even scrolling down to the bottom, while others showed quite some interactions with items they marked as skipped. Additionally, "reading" for some users meant "reading carefully", while others marked items as read that were visible for four seconds only. Nevertheless, the results should be able to point out how information on client-side interactions could be used to identify reading.

Table 1 displays the minimum, maximum, mean and standard deviation of all the recorded variables. The table shows that the mouse cursor was, on average, just a few seconds above each item or on the same y-level. The time items were visible on screen differed widely, with an average of $25.15s$ and a standard deviation of $21.14s$.

	N	min	max	mean	SD
mouse time above item	1183	0	63	3.42	5.60
mouse time on same y-level as item	1183	0	64	4.37	5.99
time item is visible in browser window	1183	0	120	25.15	21.14
amount of mouse moves above item	1183	0	30	1.20	2.64
number of mouse clicks on item	1183	0	9	0.11	0.63
number of text selections inside item	1183	0	2	0.01	0.09

Table 1: Descriptives of assessed variables

Finally, more than half of the news items (57%) were skipped, about one quarter was glanced at (23%) and 20%, on average, was read by the participants.

The first step in determining which factors influence item skipping, glancing or reading behavior was to assess the correlations among the variables. The results can be found in Table 2. It shows correlations between all the recorded mouse actions and time measurements and item skipping or reading behavior. Participants' glancing behavior is not correlated with any of the assessed variables. In other words, item glancing behavior cannot be predicted with any of the measured variables. All measured variables have a positive correlation with reading and a negative one with skipping items. This shows that they might be used to determine whether something has been read or skipped.

	A	B	C	D	E	F
item skipped	$-.28$*	$-.29$*	$-.20$*	$-.28$*	$-.17$*	$-.06$**
item glanced at	$.03$	$.04$	$.02$	$.03$	$-.01$	$.01$
item read	$.31$*	$.31$*	$.23$*	$.26$*	$.22$*	$.07$**

A) mouse time above item
B) mouse time on same y-level as item
C) time item is visible in browser window
$*$ significant at 0.01 level (2-tailed)

D) amount of mouse moves above item
E) number of mouse clicks on item
F) number of text selections inside item
$**$ significant at 0.05 level (2-tailed)

Table 2: Correlations among variables

However, the direct correlation between the assessed variables and reading behavior is not very strong, which is due to the fact that reading an item does not necessarily result in observable interactions. Nevertheless, the variables may be used as unidirectional indicators for reading behavior. One example is the selection of text. If text has been selected, the item has definitely not been skipped. However, as in 99.7% of the presented items no selection of text took place, the lack of text selections does not give any information at all.

Similarly, all assessed variables have been analyzed to find implications to be derived from the observed data. Table 3 shows how often information on client-side behavior could be retrieved and how measuring interaction times or the occurrence of interactions were related to users' responses on whether an item has been read.

	mouse over	mouse: same y	visible time	mouse moves	mouse clicks	text: select
value > 0	57.3%	74.1%	94.2%	35.6%	5.4%	0.3%

Total percentage of items where client-side data could be retrieved

	mouse over	mouse: same y	visible time	mouse moves	mouse clicks	text: select
read if 0	12.9%	12.7%	1.4%	14.0%	17.1%	19.8%
read if > 0	25.2%	22.5%	21.1%	30.6%	70.3%	75.0%
glanced if 0	21.8%	17.3%	2.9%	24.0%	23.2%	22.9%
glanced if > 0	23.7%	24.9%	24.1%	20.9%	17.2%	25.0%
skipped if 0	65.3%	69.9%	95.7%	61.9%	59.7%	57.3%
skipped if > 0	51.0%	52.7%	54.8%	48.5%	12.5%	0.0%

Total percentage of items having been read / glanced at / skipped depending on whether client-side monitoring returned a value > 0

Table 3: Occurrence of interactions

Generally it may be said that the observation of client-side interactions at least doubles the probability that an item has been read. 80% of the items with no monitored interactions or an interaction time $< 0.5s$ (rounded to 0) have not been read and most of them have been skipped.

However, half of the items where interaction times have been measured or mouse moves have been monitored have been skipped as well. Therefore, the second part of the current section consists of a closer analysis of the assessed variables and should show how higher activity values correspond to a higher probability that something has been read.

Analyzing Mouse Over Time 52.8% of all items that have been read had a mouse over time of more than 3 seconds. Items with a total mouse over time of more than 8 seconds (12.9% of all cases) have a 0.50 probability of having been read and a 0.77 probability that the item has not been skipped.

As shown in Figure 2 a higher mouse over time goes along with a higher probability that an item has been read.

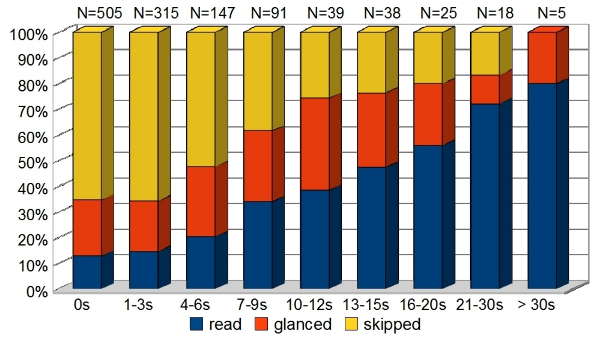


Figure 2: correlation of mouse over time and reading

Analyzing Vertical Mouse Position The time the mouse cursor has been placed at a vertical position within the borders of the news item is similar to the mouse over time and therefore the results as well (see Figure 3).

Compared to the mouse over time, the probability of the mouse never having been on the same y position as an item is lower (of course; as hovering an item implies that the mouse is also on the same vertical position). Comparing $\mathit{mouse\ over} < 1s$ and $\mathit{same\ y} < 2s$ both cover more or less the same test cases. Generally, the small differences between mouse over time and the vertical mouse position lead to the assumption that users that placed their mouse cursor inside the page tended to place it above the news items. However, this effect might have been different if more items had been placed next to each other on the same vertical position. Further work needs to be done to tell whether the y-position of the mouse or exact hovering is more significant in a different context.

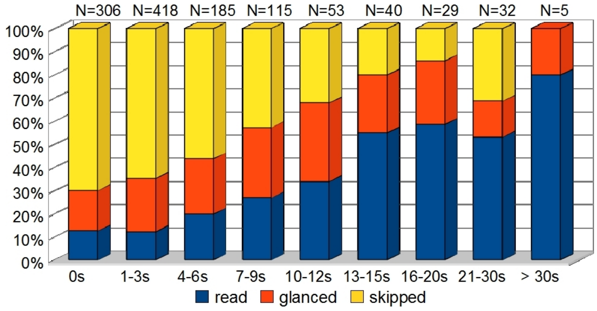


Figure 3: correlation of vertical mouse position and reading

Analyzing Visibility Time 81.1% of the items that have been visible for less than 5 seconds (13.4% of all cases) have been marked as skipped. The probability that an item has not been read (i.e. skipped or glanced at) if it has been visible for less than 5 seconds is 0.93. Only 1% of all items have been marked as read and were visible for less than 5 seconds (no surprise as items have to be visible to be read).

Other than this the visibility time does not provide any relevant information. As shown in Figure 4 the probability that an item has been read increases only slightly with a higher visibility time. This increase is definitely not sufficient for drawing further conclusions.

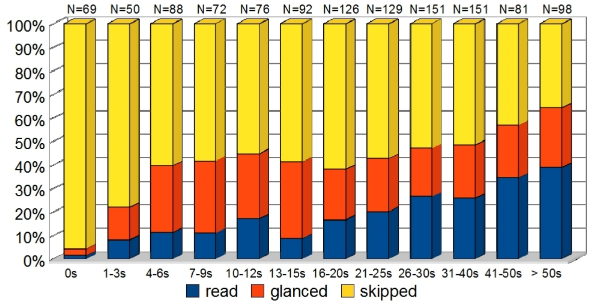


Figure 4: correlation of visibility time and reading

Nevertheless, taking into account the screen size and consequently the number of items displayed at the same time, it might have been possible to derive a weighted metric combining visible time with screen size that might have been more informative the the time by itself. Moreover, the relative position of the item within the screen might give additional information if it can be found that for instance users tend to read text that is displayed in the center of the screen. These two aspects will be considered in future experiments.

Analyzing Mouse Moves 91.1% of the skipped items had only 2 or less registered mouse moves and 98.5% of all skipped items had 5 or less registered mouse moves. Moreover, 54.8% of the items that have been read had at least one registered mouse move. No registration of mouse moves is a good indicator for having been skipped and a high amount leads to the assumption something has been read. Only 0.8% of all monitored news items have been marked as skipped despite having more than 5 mouse moves.

Detailed information can be found in Figure 5.

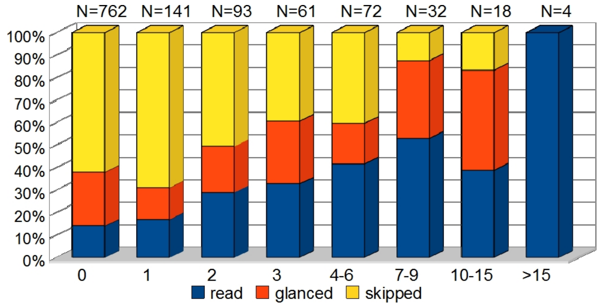


Figure 5: correlation of the number of mouse moves within a news item and reading

Analyzing Click Events Only in 5.4% of the cases clicks have been registered. However, 70.3% of them have been marked as read. Only 0.7% of the test cases showed clicks despite having been marked as skipped. This shows that although clicks do not occur frequently, they are a strong indicator that something has been read.

Analyzing Select Events As already mentioned, text selections occur even less frequently than click events (only 0.3% of all test cases). Nevertheless, text selections are the strongest indicator for reading and none of the items where text selections took place has been marked as skipped.

6 Towards an Algorithm

The results of this user study show that information on client-side user interaction is definitely suited for determining which parts of a page have been read or skipped. However, the observed variables provide different types of information. In some cases (especially interaction times) the lack of information is an indicator for skipping, in others (especially interactions) there is little probability that something can be observed, but if interactions have been monitored they serve as an indicator for reading. The visibility time works very well for identifying skipped items, but high visibility times do not really increase the probability that an item has been read (although, as discussed in the previous section, this effect might be reduced by considering the size of the browser window and the relative position of the items within). Clicks and text selections help to identify read items, but do not work for identifying skipped items.

Based on this information it may be said that when looking for an algorithm returning a probability that an item has been read, linear algorithms are definitely not the best choice. Linear models can still be informative though in terms of the viability of using specific factors and indices into the algorithm. To explore this premise we started our analysis using the following composite metric (which was only intended to give a quick impression on whether the variables might be suited to analyze reading behavior): $(1 + mouse\ \ over\ \ time) * (visible\ \ time) * (1 + mouse\ \ moves) * (1 + clicks)$. The value for the mouse on the same y position is part of the mouse over time and text selections hardly ever occurred, so these two variables have been left out. If the visible time is 0 the item can be regarded as skipped, but for all other variables even a value of 0 could mean it has been read – depending on the other variables. Therefore those variables have been added with 1. Using this simple algorithm 68% of all read items had a value above 108 and 68% of the skipped items had a value below 108. These values are of course specific to the experimental data at hand, and would in all likelihood differ significantly in other cases. However, the results do indicate that these factors do indeed have discriminatory capacity, and, possibly in an appropriately weighted form, can indeed be used as the basis for an algorithmic approach.

Having established at least some of the factors that an algorithm could incorporate, we turned our attention to the nature of the algorithm that could be used to identify page segments that had been read. The primary design requirements were:

- *real-time*: The algorithm should be fast enough to provide just-in-time information for several users while continuously monitoring user interactions.
- *predictive*: The algorithm should be able to handle continuously updating information without relying on an analysis after a user left the page.
- *white box*: The algorithm should consist of semantically understandable parts in order to be able to extend the algorithm and add factors later (or set different factors for different contexts).

Based on these requirements we decided to direct our attention to rule-based approaches, Bayesian networks and decision trees, as well as hybrid approaches comprising the above and potentially complementary ones as well.

In order to find a more appropriate algorithm the open source data mining software Weka [Witten and Frank, 2005] has been used applying different machine learning algorithms for classification. This software may be used to automatically generate models for classification algorithms by using a subset of the raw data. The other part of the raw data is used to evaluate the deriving models in order to determine how well data sets can be classified. The exact way in which the data set is split affects the performance of the algorithms. Therefore, a 100-fold cross validation has been used, i.e. the data set is randomly split 100 times and the result refers to the average value for all test cases.

The data from the user study was used to get an algorithm for predicting whether an item has been read fully or not. For the purposes of the analysis presented herein, “glanced at” and “skipped” have been combined to a single group. For each of these two classes the number of correctly classified items has been calculated as well as the precision (the probability that the item has really been read / not read, if the algorithm classified it this way).

Most algorithms had an overall precision of $\approx\ 80\%$. They showed good results especially for identifying items that have not been read. More than 95% of the “not read” items have been classified correctly (with an overall precision above 80%). However, the algorithms were less successful in identifying items that have been read. The total precision for items classified as “read” was $\approx 60\% - 70\%$ and only $15\% - 30\%$ of the read items have been correctly classified as read.

As an example three different classifiers will be discussed in details. They have all been tested using a 100-fold cross validation in Weka. The results are listed in Table 4.

One simple approach for classification is a rule based algorithm:

```
read =
   (mouse_moves >= 2 && same_y >= 13) ||
   (visible_time >= 17 && clicks >= 1)
```

The highest precision for read items was reached by the Bayesian Network shown in Figure 6. On the other hand it only classified 16% of the read items correctly.

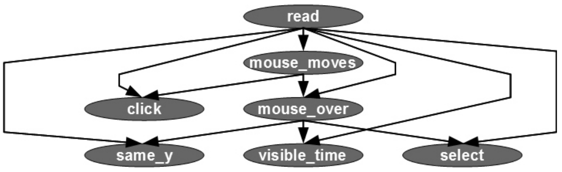


Figure 6: model for setting up a Bayesian Network

The highest average precision and the highest percentage of correctly classified read items was reached by the decision tree shown in Figure 7.

The above results clearly indicate that more work needs to be expended in devising a generic algorithm, as well as in understanding how different interaction- and context- characteristics influence the significance of the identified factors (and of how to incorporate these varying levels of significance in the algorithm itself). Nevertheless, the results seem promising in terms of being able to use client-side interactions to make assumptions on reading behavior.

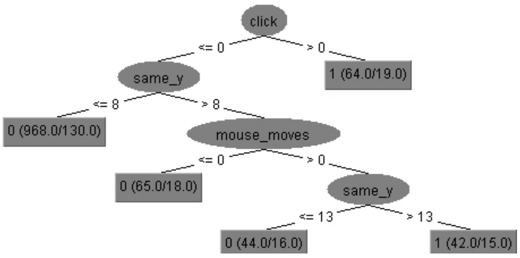


Figure 7: decision tree

	rule-based	Bayesian Net	tree-based
correctly classified read	25.8%	16.1%	30.5%
correctly classified not read	96.0%	98.4%	95.8%
correctly classified (average)	82.0%	82.0%	82.8%
precision read	61.6%	71.7%	64.3%
precision not read	83.9%	82.5%	84.7%
precision (average)	79.4%	80.3%	80.6%

Table 4: performance of different classifiers in Weka

7 Ongoing Work and Future Perspective

As a next step an experiment comparing the current work with eye-tracking will be performed. This should show how mouse positions are related to the locus of attention and whether client-side monitoring could provide parts of the information available through eye-tracking. Moreover, it should show whether users have preferences concerning the relative position of what they are currently reading (i.e. whether they focus more on elements that are displayed at the center of the screen). If this is found to be relevant, the library will be extended to monitor the relative position of page fragments on the screen and get more fine-grained information on the visibility time.

Furthermore, the library will be extended to get more fine-grained information on user behavior. This includes monitoring the scrolling speed and the size of the browser window. As the number of items visible in parallel depends on the window size (e.g.: big screen vs. mobile device), this may help to better use the visible time (fewer items on a screen increase the probability for a single one being read). Moreover, client-side monitoring should be used in different contexts; the way of reading a news page may be different from reading text in an e-learning course.

As a strong correlation between “mouse over” and “mouse on same y” has been found, it has to be tested whether this is also true if several items are placed at the same vertical position or whether it is possible to ignore information on the horizontal position of the mouse.

Another important factor for future research is the length of the text within a single item. This length is important to estimate the time required for reading. For the current experiment only elements of almost the same structure and length have been used to reduce the complexity of the test. The average reading speed as well as the estimated personal reading speed in relation to the length of the text comprised by a monitored page fragment are additional factors that we believe may need to be considered as factors and incorporated into the algorithm. Based on the estimated required reading time the visibility times and interaction times could possibly provide additional information.

The main work however lies in the further development of an algorithm (or a number of algorithms that work for different contexts). The results of this work should be integrated into a version of AHA [De Bra and Ruiter, 2001] running in the open source learning platform Sakai [Sakai, 2009] to provide the findings of ongoing research for a larger audience and help to improve existing AHS.

Acknowledgments

Part of the work reported in this paper was funded by the “Adaptive Support for Collaborative E-Learning” (ASCOLLA) project, supported by the Austrian Science Fund (FWF; project number P20260-N15).

References

[Brusilovsky, 1996] Peter Brusilovsky. Methods and Techniques of Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.

[Claypool *et al.*, 2001] Mark Claypool, Phong Le, Makoto Wased, and David Brown. Implicit interest indicators. In *Intelligent User Interfaces*, pages 33–40, 2001.

[Conati *et al.*, 2007] Cristina Conati, Christina Merten, Saleema Amershi, and Kasia Muldner. Using Eye-tracking Data for High-Level User Modeling in Adaptive Interfaces. In *Proc. of The AAAI Conference on Artificial Intelligence (AAAI 07)*, pages 1614–1617, 2007.

[De Bra and Ruiter, 2001] P. De Bra and J.-P. Ruiter. AHA! Adaptive Hypermedia for All. In *Proceedings of the WebNet Conference*, pages 262–268, October 2001.

[Farzan and Brusilovsky, 2005] R. Farzan and P. Brusilovsky. Social Navigation Support in E-Learning: What are the Real Footprints? In *Third Workshop on Intelligent Techniques for Web Personalization (ITWP ’05). At 19th Int. Joint Conf. on Artificial Intelligence*, 2005.

[Goecks and Shavlik, 2000] Jeremy Goecks and Jude W. Shavlik. Learning Users’ Interests by Unobtrusively Observing Their Normal Behavior. In *International Conference on Intelligent User Interfaces - Proceedings of the 5th international conference on Intelligent user interfaces*, pages 129–132, 2000.

[Hauger, 2008] David Hauger. Fine-grained user models by means of asynchronous web technologies. In *ABIS 2008 - Adaptivity and User Modeling in Interactive Systems*, pages 17–19, Würzburg, Germany, 2008.

[Hauger, 2009] David Hauger. Using Asynchronous Client-Side User Monitoring to Enhance User Modeling in Adaptive E-Learning Systems. In *UMAP 09: Adaptation and Personalization for Web 2.0*, 2009.

[Hijikata, 2004] Yoshinori Hijikata. Implicit user profiling for on demand relevance feedback. In *IUI ’04: Proceedings of the 9th international conference on Intelligent user interfaces*, pages 198–205, 2004.

[Hofmann *et al.*, 2006] K. Hofmann, C. Reed, and H. Holz. Unobtrusive Data Collection for Web-Based Social Navigation. In *Workshop on the Social Navigation and Community based Adaptation Technologies*, 2006.

[Putzinger, 2007] A. Putzinger. Towards Asynchronous Adaptive Hypermedia: An Unobtrusive Generic Help System. In A. Hinneburg, editor, *LWA 2007: Lernen - Wissen - Adaption*, pages 383–388, 2007.

[Sakai, 2009] Sakai. collaboration and learning for educators, by educators, free and open source. http://sakaiproject.org, 2009.

[Witten and Frank, 2005] Ian H. Witten and Eibe Frank. *Data mining: practical machine learning tools and techniques*. San Francisco, 2nd edition, 2005.

Unstructured Interaction: Integrating informal handwritten knowledge into Business Processes

Felix Heinrichs
Darmstadt University of Technology
Hochschulstr. 10, D-64289 Darmstadt, Germany
+49. (0) 6151 16 6670
felix_h@tk.informatik.tu-darmstadt.de

Abstract

Business processes are a widespread approach to managing and planning organizational activities. However, human work practices often differ from structured, formal process descriptions. Knowledge on process variations therefore becomes a key aspect in enacting and controlling business processes. Such knowledge usually is informally documented. Traditional paper, carrying handwritten information, still serves as one of the prevalent media in this context. Even in computer supported work environments, paper based documents are still common in professional settings. As a solution to the problem of integrating handwritten, informally specified knowledge into business processes, the concept of *unstructured interaction* is introduced and elaborated.

1 Introduction

Contemporary Business Process Management (BPM) solutions rely upon formal process models. On the one hand, formal process models alleviate the mapping from process to information technology, thereby providing a huge potential for efficiency gains. On the other hand, they impose severe constraints for human actors involved in the process. In reality, people tend to vary the routines, make concessions and negotiate informally. These work practices should actively be supported by business process management solutions [18]. Current systems, however, offer no satisfactory support for such practices.

People document knowledge on process variations in an unstructured, informal way, e.g. as notes, post-its or sketches. Paper in general and handwritten documents in particular – in computer supported work environments complemented by electronic notes, memos and even emails – are still common in this context, due to their flexibility and ease-of-use [14].

Looking at business processes and the role a human actor plays in these processes from an organizational perspective, allows for easier management of an organization's processes. In reality, however, a single person is simultaneously involved in various processes, she even plays several roles in a single process. This requires a new modeling perspective, focusing on the individual involvement in processes [5].

As a result, knowledge documented by individuals is subject to the very same conditions: A post-it on an employee's desktop could convey information on any of the current processes this employee is involved in. Information might be spread on several separate locations and media, jointly representing knowledge relevant to a single business process, or a combination of the processes a person is involved in.

This leads to the question: How could the process management solution benefit from the informally specified, heterogeneously documented knowledge acquired during the process of regular human work?

As solution, the concept of *unstructured interaction* is introduced. Conveying information to contribute knowledge to a business process is regarded here as a form of *interaction* with the process itself. There are no constraints on the contents, form or location of information, so it is *unstructured* in its nature. This *unstructured interaction* can be decoupled from the current state of the process and the medium used to interact with and subsequently be reintegrated into the process, using process and user context models combined with a semantic analysis of the contents of information. It is therefore also *unstructured* regarding to the structured process.

2 Scenario

To illustrate the concept of unstructured interaction in business processes, consider the example of Sally and Tom. Tom is an employee working as floor man in a large grocery store. Sally is a customer in the very same store. From the perspective of the store keeper, both participate in the business process of selling goods. Sally in the role of an external actor (customer) and Tom as an internal actor (employee).

Sally is a working mother of three children, she has only limited time resources for grocery shopping and needs to plan her tours. As a result, Sally relies on a handwritten grocery list to organize her shopping. She tried organizing these shopping tours on her smart phone, but eventually fell back to the handwritten list which allows for easier and more natural planning. Today, Sally enters her favorite store and discovers an information sign telling her of a new digital grocery shopping assistant offered by the store.

Curious, Sally reads more information about the program. It turns out that the store offers an application Sally might load on her smart phone, capable of aiding Sally's shopping tour by revealing the location of goods and doing some approximate pricing calculation for her. This all will be performed based on her grocery list. In turn, the store requires the permission to evaluate Sally's anonymized shopping behavior and her grocery list. The only requirement for participation is the possession of a smart phone and a digital pen. Luckily Sally has both.

She decides to use the added value application offered by the store. To communicate this, she confirms it on her smartphone. As Sally already wrote her grocery list using her digital pen, she can start using the new application immediately. Behind the scenes, the data on Sally's digital pen (her grocery list) is transferred to the smartphone. She simply starts shopping as usual.

Meanwhile, Tom started filling the store's racks with items. When he went to the storage room to refill the milk stock in the store, he discovered that the milk had accidentally been delivered after the best-before date had expired. The pallet actually showed a different best-before date than the individual cardboards. Tom decides to leave the milk in the storage room, as it cannot be sold anyway. He marks the damage on the checklist form he uses to track his work when filling the stores racks. The checklist, has no special rubric called ”damages” or even ”comments”. So Tom crosses out the milk, and writes a short note besides it (“Milk tainted”).

While Tom was busy in the storage room, Sally continued her shopping tour in the store. Eventually she approaches the rack supposed to contain the milk. Sally notes that there is no milk left on the rack, although she intended to buy some and therefore wrote it on her grocery list. So Sally notes the missing milk on her grocery list with a small cross-sign, differentiating it from the other items she found.

At this point, Sally returns home as an unsatisfied customer, while Tom proceeds with filling the racks in the store. Eventually, after Tom finished his tour, he reports the situation to the store manager. The store manager files a complaint for the damage and orders another milk pallet. This pallet is delivered the next morning, because of the evening delivery truck just having left its station.

If the handwritten annotations could have been processed as unstructured interaction, the business process management system would have been able to react more quickly. Tom's annotations would have been processed immediately and forwarded directly to the store manager. She would have been able to order the new delivery for the same evening. Sally's intelligent shopping assistant could acquire the information of Sally's desire for milk, which is currently unsatisfiable. So it could present her with alternative suggestions through her smart phone, for example directions to the next store, or the time milk will be available again in this particular store.

3 Related Work

Related work essentially falls into either of the following three categories: Business Process Management in general, the role paper documents play in current business processes and pen and paper based interaction between a person and a digital system.

3.1 Business Process Management

In the 1990's, the concept of business processes began to appear in literature and research [20] to describe and manage organizational activities or even construct complete business-process-centric organizations. Although no common agreement on the definition of the term *Business Process* exists due to different possible levels of abstraction, empirical studies indicate that the understanding of business process in industry involves three perspectives: i) structured processes allowing for easier management, ii) methodologies to achieve business goals and iii) sociotechnical constructs with a focus on human interactions and relationships [20]. As a result of their aforementioned studies, Vergidis, Turner and Tiwari point out that a strong preference towards the first perspective exists amongst queried companies.

A widely accepted definition of *Business Process Management* (BPM) has been provided by van der Aalst, Hofstede and Weske. They defined it as

> *Supporting business processes using methods, techniques and software to design, enact, control and analyze operational processes involving humans, organizations, applications, documents and other sources of information* [19]

Based on this definition they described a business process management system as a software system driven by explicit process designs serving to enact and manage operational business processes [19].

A common problem that business process management systems have to deal with, are variations. People do not always follow structured routines, they tend to vary such routines when encountering exceptional conditions. However, tracking or even managing variations in a business process management system proves to be challenging. Recent approaches therefore focus on bottom-up concepts, shifting the focus from the organizational to the individual perspective of users (e.g. [18], [12]).

An individual perspective allows for taking the interaction between persons and business processes into account. Genovese, Comport and Hayward described the changes of processes based on the actions of process actors as person to process interaction [5] and emphasized the need for such concepts in BPM systems. Their informal definition of person to process interaction served as basis for the view taken in the presented approach.

3.2 Paper as Medium in Business Processes

Writing on paper essentially serves conveying information, either focused on temporal (e.g. documentation), spatial (e.g. writing a letter) or social (e.g. communication, collaboration) aspects. With traditional paper still being a prevalent medium in business processes [14], conveying written information is an essential part of current work practices. Professionals, especially knowledge workers, tend to use handwriting for informal note taking [4].

Traditional paper affordances are numerous. Paper provides a very robust, flexible, mobile and cheap medium compared to most digital systems [14], allowing instantaneous interaction without annoying start-up times [21]. However, digital systems offer also clear advantages when it comes to information management, hyperlinking, communication etc. least to speak of processing and computation capabilities. Integrating both worlds, bridging the gap between the physical information documented on traditional paper and the information stored and managed in digital systems, has been the goal of many approaches (e.g. [21], [17], [6], [10]).

3.3 Pen and Paper Based Interaction

Pen and paper based interaction (PPI) describes a form of interaction between a user and a digital system using paper and a *digital pen* as input media. A digital pen essentially is an ordinary pen capable of tracking its movements either in relation to other media (e.g. paper or a display device) or in an absolute fashion. Whether a user actual inks the document, i.e. physically alters the structure of paper by letting ink colorize certain locations, depends on the type of interaction. Gesturing on paper without inking also falls in the category of PPI. An empirically validated set of basic interactions between users and digital systems through usage of paper and digital pens can be found in [17]. Additionally, a set of potential usage scenarios for pen and paper based interaction techniques is described in [4].

The Anoto Pen

Digital pens facilitate pen and paper based interaction. A prominent example for such a digital pen is the Anoto pen [1]. It is employed in most current PPI based user interfaces ([21], [17], [6], [10], [16]). To track pen movements, this technology relies on a proprietary dot-pattern. The pattern is printed onto traditional paper. A camera built into the Anoto pen scans the page for this dot pattern. Based on the scanned dots, the absolute position of the pen on paper can be determined. It is even possible to uniquely identify the page the pen is moving on.

Anoto based interaction allows more than conveying information on paper through inking, which then is transferred into a digital system. Gesture systems have been built to allow control of digital functionality thus narrowing the gap between the physical and digital world even further (e.g. [8], [17]). However, a problem special to pen and paper based interaction is the feedback channel. While the pen itself potentially leaves physical marks on paper by inking it, providing digital feedback requires additional concepts. To compensate this, Liao, Guimbretière and Loeckenhoff designed a prototypical extension to the digital pen, which is capable to provide visual, acoustic and haptic feedback [9]. Newer digital pens also employ similar feedback mechanisms [13].

3.4 Summary

If a business process management system supports pen and paper based interaction, information conveyed on paper could be integrated into the process using *unstructured interaction*. Beaudouin-Lafon described two essential levels for analysis and design of interaction between a digital system and a person [3]. *Interaction paradigms* provide a user centered high-level conception of the phenomenon of interaction, while *interaction models* offer operational descriptions of the course of interaction. Interaction models, such as for example *instrumental interaction* [2] or *direct manipulation* [15], provide guidelines for interaction design.

Reality based interaction provides a conceptual framework on a higher layer of abstraction [7], a specialized view on interaction between people and digital systems designed to conceptualize evolving interaction techniques. However, none of these approaches is suitable to describe the concept of unstructured interaction as presented in this paper.

4 Unstructured Interaction

Unstructured interaction essentially provides a different view on the course of interaction than traditional approaches. It differs mainly by not assuming that there is an underlying structure for interaction, a common language both interacting entities are required to understand completely. Instead it assumes that one entity acts and the other entity reacts based on its interpretation of this action. In the following, this concept is elaborated on with the goal to derive a definition of unstructured interaction by discrimination.

4.1 Structured Interaction

Interaction describes a phenomenon where the actions of two entities mutually affect each other. In case of interaction between a person, referred to as the *user*, and a digital system, the actions of the user affect the internal state of the digital system, while the perceived system state affects the actions of the user. The hardware and software components allowing the user to interact with the digital system are commonly referred to as the *user interface* (UI).

Traditional UIs, such as the widely known graphical user interfaces (GUI) predominantly employed in today's computer systems, restrict the possible way's of interaction. Only a limited set of control actions is supported and their affordances are exposed by the system. The UI thus ideally provides a set of necessary and sufficient actions to control the system, a control language the user needs to understand. In the following, interaction based on such a concept is referred to as *Structured Interaction*. An example for UIs based on structured interaction are today's (still) predominant WIMP interfaces (Window, Icon, Menu, Pointer).

A drawback of such an approach is that only those actions the UI designers considered are supported. Furthermore, mechanisms to distinguish valid actions from invalid ones are needed. Designers need to pay attention that the affordances communicated by the system and controls possible through its UI match (c.f. Norman's *Gulf of Execution*[11]) and that no illegal states are possible.

4.2 Unstructured Interaction

Unstructured Interaction takes an alternative approach towards the problem. No structuring of the interface by active restriction is presumed. Users might express their intentions only limited by the physical constraints of the user interface: the system strives to understand the actions of the user and interprets them. UIs based on natural language processing provide an example for user interfaces of this category. The problems for the designer shift form restriction and affordance control, to understanding. This leads to our definition of unstructured interaction:

> *Unstructured interaction describes interaction which is not based on the underlying structure of a formal language completely understood by interacting entities. Interacting entities strive to understand each other by interpretation, resembling the informal form of human communication.*

Following Beaudouin-Lafon's approach to analyze the phenomenon of human computer interaction [3], the principle of unstructured interaction encourages the *computer-as-partner* interaction paradigm. In such an interaction paradigm, the user delegates tasks to the computer. However, UIs designed to support unstructured interaction might also be employed in systems supporting different interactions paradigms (i.e. *computer-as-tool*, *computer-as-medium*).

4.3 Unstructured Interaction in Business Processes

Bringing information conveyed on physical documents into intelligent business processes essentially forms an interaction between a person and a process. The person, with respect to the process here referred to as *actor*, conveys some information relevant to the process. The process reacts to

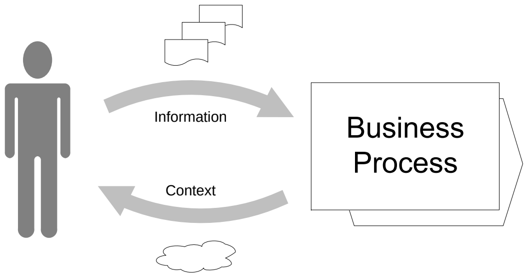


Figure 1: Interaction between persons and business processes

this information, meaning that its further flow of execution depends on the information. This in turn affects the actor by changing her context, as she is playing a role in the process. Consequently, the actions of two entities mutually affect each other, as shown in Figure 1 and we can speak of *interaction*. Following the terminology introduced in [5], such interaction is here referred to as *person to process interaction*. BPM systems designed to support intelligent business processes have also to provide support for person to process interaction: an appropriate user interface is needed.

Using structured interaction in person to process interaction means that the information conveyed needs to be formalized. However, formalization contradicts the informal way information is gathered and documented throughout the process of work and results in loosing part of the information. Only those interactions the system designer considered beforehand are allowed, which is by definition not the case under exceptional conditions.

Person to process interaction based on the concept of unstructured interaction contributes to the solution of such problems. User actions are interpreted by the system. It analyzes information obtained from user actions, generates knowledge on the current situation and re-integrates this knowledge into the managed process. Although the system itself is unable to process information it cannot understand, the borders which information potentially could be included at which point in time would become softer.

4.4 Pen and Paper based Interaction (PPI) in Business Processes

The absence of a formal structuring for the contents of blank paper ("what" can be written on it) serves the need to convey *any* desired information. For example, engineers might convey information consisting of a mixture of mathematical formulae, technical drawings and written sentences to document a specific design idea. A text entry field on their computer or smartphone simply does not allow to do so, unlike an empty sheet of paper.

Recent studies corroborate this. Chapman, Lahav and Burgess report in their field study on handwritten documentation practices in enterprises a free mix of text, drawings, mathematical symbols and drug or term abbreviations, and a distribution of information on many, sometimes casual, paper artifacts [4].

Paper is commonly used in current work practices [14]. A lot of information important to business processes resides on paper artifacts. Thus it cannot be accessed by the business process management systems employed. Information conveyed on paper artifacts is heterogeneously documented, informal and unstructured. Hence it can be concluded that pen and paper based interaction between a person and process needs a user interface based on an unstructured interaction concept, rather than limiting the potential by artificially structuring the interaction. The system needs to understand as much as possible of the information conveyed by the user, without the guarantee to understand the complete information.

5 Support of Unstructured Interaction in Business Processes

Granted that interfaces based on the concept of unstructured interaction contribute to the solution of integrating knowledge in business processes, the question remains how to realize such interfaces. How would actual system support be realized? How could the sheer complexity of informally specified information be handled?

The central idea to reduce complexity is to take the context of interaction into account. Formal process descriptions of business processes being executed provide such structured context. It can be assumed, that the current tasks an individual person is involved in are available. Based on these individual tasks, the interaction itself can be interpreted and related to the ongoing activities. So the formal process description provides a structural framework for informal information.

5.1 The Interaction Processing Pipeline

As shown in section 4.4, pen and paper based interaction (PPI) provides an example of unstructured interaction in the domain of business processes. Figure 2 displays the relevant aspects of unstructured interaction support for pen and paper based interaction. As you can see, realizing unstructured PPI involves the following steps

(i) Process

(ii) Understand

(iii) Integrate

Processing describes the conversion of raw inking or gesture data to meaningful constructs. Samples of pen positions are transformed into strokes and gestures. Strokes and pen movements on a document are segmented into areas containing drawings, written information, formulae etc. Knowledge on the process context could already be used at

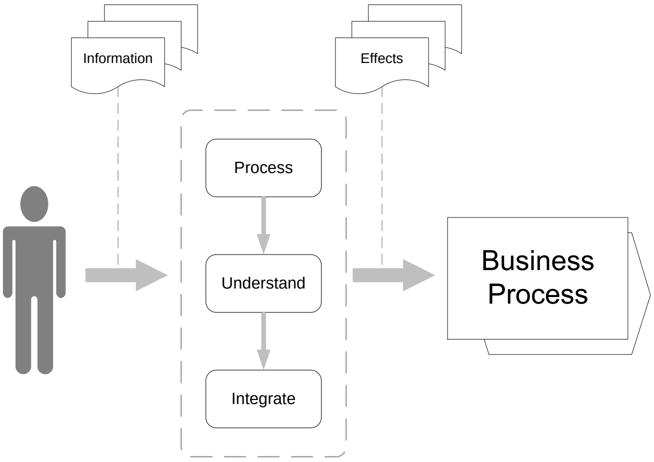


Figure 2: The interaction processing pipeline for pen and paper based interaction in business processes

the processing stage to identify written objects. For example, if a sales agent inks a note while she discusses something with a client on the phone, the information might more likely be text than formulae.

Understanding takes the processed information and interprets it. Information is placed in the context of current processes and a semantic model of this information and its relation to a current process is derived. Then the appropriate processes to which this information might be relevant are identified. Naturally, this step also relies on the current process context.

Integrate finally uses the semantic information model and the process context to integrate the information into the process. The general process model and its currently running instance are analyzed and the optimal type of integration is determined. Such integration might for example be choosing an alternative flow, firing a process event or altering a process artifact (i.e. a document).

A business process management system supporting unstructured PPI will have to address these issues. It needs to employ an information or rather interaction processing pipeline and adequate business process context models.

5.2 User Based Perspective

It has been shown, that a user centric perspective in business process management based on the work of individuals involved in process provides several benefits [18], [12], [5]. Following a user centric approach to realize person to process interaction based on the concept of unstructured interaction therefore requires the business process management system to take the multiple roles a user might play in several business processes into account. The motivation and rationale behind user participation in the business process is another relevant aspect contributing to context information.

Essentially, users, or to be precise *actors*, in business processes could be grouped based on their motivation or goals into

(i) internal actors

(ii) external actors

Internal actors are entities carrying out work activities with the goal to advance the business process. Incremental advancement ultimately results in the completion of a business process. In typical constellations such actors are employees or systems of the organization executing the business process, although exceptions are thinkable. Goals of internal actors are related to the goals of the business process as a whole generally speaking. On the contrary, *External actors* are entities carrying out activities based on goals unrelated to the goals of the business process. Such actors are in most cases customers or external systems.

Interaction between the process and a user depends on the role this user assumes. Consider internal actors, which contribute to the advancement of a business process intentionally. Therefore the state of a current process instance provides a strong indicator for processing of information written by internal actors.

External actors might also convey written information. In their case, the state of an organizations business processes does not necessarily provide an indicator for the context of written information. However, compared to information conveyed by internal actors, their information might not be as relevant regarding the business process.

Based on the user role, two complementary use cases for unstructured pen and paper based interaction in business processes can be identified. They are illustrated in the scenario in section 2. Although this selection of use cases is far from complete, it illustrates usage of the same concept in two complementary applications.

The first use case shows how internal actors annotate documents with information relevant to the process (c.f. Tom and the comments he made). The second use case describes integrating informally specified information from external sources (c.f. Sally and the grocery list). Both are merged into a single scenario, to illustrate the interconnection between informations and several simultaneously executed processes (c.f. the supermarket's process and Sally's personal shopping process).

6 Conclusion and Research Perspective

The contributions of this paper are twofold. First, the concept of *unstructured interaction* was defined and elaborated. Its importance and applicability in integrating handwritten information into business processes was highlighted. Second, an initial guideline for the design of business process management solutions based on such an interaction concept has been proposed. Relevant processing stages for handwritten information in business processes have been pointed out. Such processing bases upon the structural framework provided by a formal process description employed in the business process management system.

The key questions for further research on the subject are oriented on the three aspects introduced in section 5. i) Concepts to process and model information conveyed on paper are needed. Even at this early stage the role of the process context and its potential impact has to be researched. ii) The question how to understand such information in the process context needs to be addressed. iii) An adequate approach to integrate this information into the business process will be needed.

References

[1] Anoto. Digital pen and paper technology. http://www.anoto.com/.

[2] Michel Beaudouin-Lafon. Instrumental interaction: an interaction model for designing post-wimp user interfaces. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 446–453, New York, NY, USA, 2000. ACM.

[3] Michel Beaudouin-Lafon. Designing interaction, not interfaces. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 15–22, New York, NY, USA, 2004. ACM.

[4] C.N. Chapman, M. Lahav, and S. Burgess. Digital pen: Four rounds of ethnographic and field research. In *Proceedings of the 42nd Hawaii International Conference on System Sciences, HICSS '09*, pages 1–10, Jan. 2009.

[5] Yvonne Genovese, Jeff Comport, and Simon Hayward. Person-to-process interaction emerges as the 'process of me'. Gartner Research, May 2006.

[6] François Guimbretière. Paper augmented digital documents. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 51–60, New York, NY, USA, 2003. ACM.

[7] Robert J.K. Jacob, Audrey Girouard, Leanne M. Hirshfield, Michael S. Horn, Orit Shaer, Erin Treacy Solovey, and Jamie Zigelbaum. Reality-based interaction: a framework for post-wimp interfaces. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 201–210, New York, NY, USA, 2008. ACM.

[8] Chunyuan Liao, François Guimbretière, Ken Hinckley, and Jim Hollan. Papiercraft: A gesture-based command system for interactive paper. *ACM Trans. Comput.-Hum. Interact.*, 14(4):1–27, 2008.

[9] Chunyuan Liao, François Guimbretière, and Corinna E. Loeckenhoff. Pen-top feedback for paper-based interfaces. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 201–210, New York, NY, USA, 2006. ACM.

[10] Wendy E. Mackay, Guillaume Pothier, Catherine Letondal, Kaare Bøegh, and Hans Erik Sørensen. The missing link: augmenting biology laboratory notebooks. In *UIST '02: Proceedings of the 15th annual ACM symposium on User interface software and technology*, pages 41–50, New York, NY, USA, 2002. ACM.

[11] Donald A. Norman. *The design of everyday things*. Basic Books, [New York], 1. basic paperback ed., [nachdr.] edition, 2002.

[12] U.V. Riss, A. Rickayzen, H. Maus, and W.M. van der Aalst. Challenges for business process and task management. *Journal of Universal Knowledge Management*, 0(2):77–100, 2005.

[13] K. Schreiner. Uniting the paper and digital worlds. *Computer Graphics and Applications, IEEE*, 28(6):6–10, Nov.-Dec. 2008.

[14] Abigail J. Sellen and Richard H. R. Harper. *The Myth of the Paperless Office*. MIT Press, Cambridge, MA, USA, 2003.

[15] B. Shneiderman. Direct manipulation: A step beyond programming languages. *Computer*, 16(8):57–69, Aug. 1983.

[16] Beat Signer and Moira C. Norrie. Paperpoint: a paper-based presentation and interactive paper prototyping tool. In *TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 57–64, New York, NY, USA, 2007. ACM.

[17] Jürgen Steimle. Designing pen-and-paper user interfaces for interaction with documents. In *TEI '09: Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, pages 197–204, New York, NY, USA, 2009. ACM.

[18] Todor Stoitsev, Stefan Scheidl, and Michael Spahn. Task models and diagrams for user interface design. In *Task Models and Diagrams for User Interface Design*, volume 4849 of *Lecture Notes in Computer Science*, pages 213 – 226, 2007.

[19] Wil van der Aalst, Arthur ter Hofstede, and Mathias Weske. Business process management: A survey. In *Business Process Management*, volume 2678/2003 of *Lecture Notes in Computer Science*, pages 1019–1019. Springer Berlin / Heidelberg, 2003.

[20] K. Vergidis, C.J. Turner, and A. Tiwari. Business process perspectives: Theoretical developments vs. real-world practice. *International Journal of Production Economics*, 114(1):91 – 104, 2008. Special Section on Competitive Advantage through Global Supply Chains.

[21] Ron Yeh, Chunyuan Liao, Scott Klemmer, François Guimbretière, Brian Lee, Boyko Kakaradov, Jeannie Stamberger, and Andreas Paepcke. Butterflynet: a mobile capture and access system for field biology research. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 571–580, New York, NY, USA, 2006. ACM.

Integrating Semantic Web and Web 2.0 Technologies for supporting Collaboration Engineering

Stefan Werner Knoll, Ernesto William De Luca,
Graham Horton, Andreas Nürnberger
Otto-von-Guericke-University of Magdeburg
Faculty of Computer Science
Universitätsplatz 2, 39106 Magdeburg, Germany
sknoll@sim-md.de, ernesto.deluca@ovgu.de,
graham@sim-md.de, andreas.nuernberger@ovgu.de

Abstract

In this paper we present ongoing research about a supporting framework that improves Group Support Systems (GSS) for Collaboration Engineering (CE). CE uses a pattern design approach which allows groups with no design skills, and only limited facilitation skills, to design and execute efficient and effective collaboration processes. Our research tries to use Web-based GSS technologies to support the CE approach. We assume that Social and Semantic Web technologies could enhance CE providing relational (formal rules) and shared information (community experiences). This leads to a new GSS approach that supports groups in designing and executing a collaboration process considering users contributions for a structured collective knowledge sharing process.

1 Introduction

Collaboration is a social and interactive process, where participants join efforts toward a group goal. The outcomes of a collaboration process can be affected by individual, group-related, organizational and social factors.

1.1 Collaboration Science

Collaboration Science studies these factors and develops concepts and methods that support collaboration work by assisting a group in combining the potential and expertise of the participants. A technical support is represented by a Group Support System (GSS) that offers a variety of tools that link a group via computers and assists them in structuring activities and improving communication [Nunamaker *et al.*, 1991]. This group-oriented framework allows the participants to work collaboratively toward a goal via the web by sharing and creating information simultaneously. However, research indicates that facilitation is a key success factor in the use of GSS [Nunamaker *et al.*, 1997], as it is difficult for groups to appropriate the GSS technology by themselves. Experience is necessary for the design of a collaboration process, its implementation in a GSS, and the facilitation during the process itself. For this reason, organisations use professional facilitators, who are experts in the design and execution of collaboration work and can improve group productivity. Skilled facilitators can be expensive and as a result most organisations cannot benefit from facilitation intervention [Briggs *et al.*, 2003]. The challenge is to find a way to design and execute an efficient and effective GSS-based collaboration process, which places no design skills, and low demands on the facilitation skills of the group.

1.2 Collaboration Engineering (CE)

Collaboration Engineering (CE) is an approach to designing and deploying collaboration processes for high value tasks which are a frequent part of routine work practices of an organisation [De Vreede *et al.*, 2005]. An example is the innovation process which creates substantial value for organizational stakeholders. CE intends to enable an organisation to increase the quality of collaboration for this kind of recurring mission critical task in the organisation. The premise of this approach is that for recurring collaboration processes, ongoing professional facilitation support is too expensive. Therefore, the preparation and design is done by a collaboration engineer (expert facilitator). The resulting design can be transferred to a practitioner (a domain expert in the organization), who should be able to guide a group of participants to achieve its goal in a satisfying way. The requirements for the collaboration process design, are recognized by the practitioner who has no deep knowledge in designing the process, and only limited facilitation skills. Therefore, CE uses a pattern approach to subdivide and classify collaboration processes into five key patterns of collaboration, which together form a pattern language for group collaboration [De Vreede *et al.*, 2005]:

1. ***Diverge**: Move from having fewer to having more concepts.*
2. ***Converge**: Move from having many concepts to a focus on and understanding of a few deemed worthy of further attention.*
3. ***Organize**: Move from less to more understanding of the relationships among concepts.*
4. ***Evaluate**: Move from less to more understanding of the benefit of concepts toward attaining a goal relative to one or more criteria.*
5. ***Build Consensus**: Move from less to more agreement among stakeholders so that they can arrive at mutually acceptable commitments.*

These collaboration processes (called thinkLets) are “named, packaged facilitation interventions that create a predictable, repeatable pattern of collaboration among people working together toward a goal” [De Vreede *et al.*, 2005]. Research has shown that practitioners who know the specification of a thinkLet can predictably and repeatable engender the pattern of collaboration a given thinkLet is intended for, even without any facilitation expertise [De Vreede *et al.*, 2005]. We assume that these properties of CE could be improved with a Web-based GSS technology. The main challenge in our research is therefore to develop a conceptual design that implements an instance of a collaboration process with GSS and provides design guidelines for an appropriate facilitation support by practitioners with limited facilitation skills.

GSS research requests to reduce the cognitive costs of searching for, assimilating, and remembering the information as well as to create a common ground for interaction among several cultures for global teams [Nunamaker *et al.*, 1991]. We think that from a practical perspective, this condition can be considered not only for the execution of a collaboration process, but also for its design. Therefore, we assume that a combination of CE with Semantic Web (for the formalization of the information representation) and Web 2.0 (for sharing information between groups and members) applications is an appropriate way of adapting information for supporting the design and execution of these collaboration processes. This combination would minimize conceptual load for inexperienced collaboration engineers and practitioners.

2 Current Research: A Web GSS Framework

Before discussing the idea of combining a collaboration framework with Social and Semantic Web applications, we first want to introduce our approach that is a Web-based GSS framework. We use the object-oriented approach of a thinkLet to create a Group Process Modeling Language (GPML) [Knoll *et al.*, 2008] that implement the pattern design approach of CE with a GSS by describing the data of collaboration processes in a compact representation. These reusable collaboration processes can be flexibly adapted for different contexts. The GPML uses the element thinkLet as a process template to create the collaboration patterns [De Vreede *et al.*, 2005] explained above. These patterns illustrate information about the order and type of the activities of a participant, the type and the value of the data elements that can be used and the influence of events on the collaboration process. By configuring the process information a process template can be adapted to a group goal. In this case, the defined activities of the thinkLet will be adapted. The GPML divides a collaboration pattern into repeatable atomic activities (like *to create a new contribution, to select a contribution from a list of contributions* or *to read information about the group task*) which represent a template of a user interface for an atomic activity of a participant [Knoll *et al.*, 2008]. By using atomic activities, the GPML can define personalized processes for the participants of a group. In this case the GPML differentiate between a sequence of activities of an individual participant and a group of participants which illustrates concurrent processes of participants with different roles.

Our first application of the GPML is a Web-based GSS for the first stages of an innovation process (the generation and selection of ideas) that links a group via the Internet and implements the activities of a collaboration process via a website [Knoll *et al.*, 2009]. This prototype is based on a framework that develops data structures and functionalities for design, execution and data management of a collaboration process. The GSS supports asynchronous communication, anonymous contribution and group-wide access to all entered contributions. Currently, the GSS uses text data as the medium for communication and presentation of information, stimuli, ideas and decisions. Processes like idea generation, clustering, selection and decision making are implemented as different collaboration processes which can be adapted and combined to the first stages of an innovation process. XML is used to define the elements of the GPML and the related configuration. The prototype provides functionalities to upload and store different templates into a library of collaboration processes. A practitioner can select a collaboration process and configure the GPML elements to a given group setting. However, the possibility of creating a “collaboration process” is static. In this case, the collaboration engineer and practitioner has no support at all, so that he/she has to decide, which elements have to be connected with another or how a process has to be adapt, trusting only his/her experience. For this reason, we think that the embedding or linking GPML to web based concepts – either from ontologies or dynamic Web 2.0 tools or services – would strongly extend the flexibility of collaboration processes.

3 Future Research: A Social and Semantic Web-based GSS Framework

Instead of only proposing a classic *static* access to the information, where a word is used to only express one concept, without taking into account its context, we want to use an ontology model for multi-criteria access [De Luca and Nürnberger, 2009]. An ontology is a formal specification of a conceptualization of a domain of interest [Gruber, 1993] that specifies a set of constraints that declare what should necessarily hold in any possible world. Ontologies are used to identify what “is” or “can be” in the world. It is the intention to build a complete world model for describing the semantics of information exchange, which nicely fits the needs of the GPML defined collaboration processes [Knoll *et al.*, 2008], where ontologies could be used to facilitate knowledge sharing and reuse. In this case, we want to enable collaboration engineers to use ontologies for creating and combining collaboration process templates. Here, a collaboration engineer can use a concept, its properties and the relationships between the concepts for creating new “configuration rules” that can be applied to the collaboration process. (For example the concepts atomic activities: (a) *to create a new contribution* and (b) *to explain the group goal* and the concept social group factor: (c) *to reduce production blocking* can be combined to the configuration rule: to support *c* use the sequence *b, a*.) Therefore, ontology for a collaboration process template will define selection and design rules for the participant's activities in connection to different group and meeting structures. We decided to redefine the elements of the GPML and their relations included in "configuration rules" with RDF/OWL expressions (http://www.w3.org/TR/owl-ref/). This syntax should support a collaboration engineer in creating a template of collaboration processes for its execution with the Web-based GSS.

The resulting templates developed by the collaboration engineers will be made available for the practitioners, who can adapt them for specific groups and meeting structures. The selection process could be supported by tagging the template of a collaboration process with different parameters like *the tasks, the goals, the group size* and *the process time*. Also the collaborative Web-based GSS could help the practitioner in giving roles to the participants, specifying different individual activities. In addition, the practitioner could add information like *the motivation*, *the relationships between the participants* and *the professional level* to better schedule every individual participant in the group. In this way, the collaboration process is accelerated, adapted for different needs, and the practitioner would benefit from the changes and the new information produced by their and different groups. In order to integrate such resources with a Social and Semantic Web-based GSS implementation, we think that a form to define these requirements should be provided as well as a selection of different process templates that fit every single practitioner request.

We assume that the existing GSS could be improved with Social Web functionalities that influence the social factors such as *distractions, production blocking, social loafing* and *evaluation apprehension*. For example, during the collaboration process, a group could use feedback or competition tools to reduce social loafing. By connecting these technologies as applications for a GSS and defining rules for their use adapted to the collaboration pattern, the participants could be supported in creating and sharing information and ideas for collaboration work. These rules could consider parameters like *the time estimation*, *the conceptual ideas* and *the related results* of a thinkLet. These properties could be retrieved from already existing ontologies, browsed by all participants and integrated in the collaborative work, so that a real dynamic collaboration process could take place.

4 Conclusions

In this paper, we discussed the possible integration of Social and Semantic Web technologies with Collaboration Engineering and Group Support System. These applications could be used for supporting groups in designing and executing a collaboration work. We showed different ideas of integration discussing how our framework for a Group Support System would allow users to access processes in a collaborative way, sharing experiences and solutions with the newest Social and Semantic Web technologies. In the future work, we will include different other scenarios, analyzing more deeply how ontologies and Web 2.0 applications have to be implemented for GSS.

References

[Briggs *et al.*, 1999] Robert O. Briggs, Mark Adkins, Daniel D. Mittleman, John Kruse, Scot Miller, and Jay F. Nunamaker Jr. A technology transition model derived from field investigation of GSS use aboard the USS Coronado, *Journal of Management Information Systems*, 15(3):151-195, December 1999.

[Briggs *et al.*, 2003] Robert O. Briggs, Gert-Jan De Vreede, and Jay F. Nunamaker Jr. Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems, *Journal of Management Information Systems*, 19(4):31-64, Spring 2003.

[De Luca and Nürnberger, 2009] Ernesto W. De Luca and Andreas Nürnberger. The RDF/OWL LexiRes Tool: Maintaining Multilingual Resources. *GSCL Journal, Special Issue GSCL-Forum - Lexical-semantic and ontological resources*, 2009.

[De Vreede *et al.*, 2005] Gert-Jan De Vreede and Robert O. Briggs. Collaboration engineering: designing repeatable processes for high-value collaborative tasks. In *Proceedings of the 38th Annual Hawaii International Conference on System Science, HICCS-38*, Big Island, Hawaii, January 2005. IEEE Computer Society.

[Gruber, 1993] Thomas R. Gruber. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199-220, 1993.

[Knoll *et al.*, 2008] Stefan W. Knoll, Martin Hörning and Graham Horton. A Design Approach for a Universal Group Support System using ThinkLets and ThinXels. In *Proceedings of the Group Decision and Negotiation 2008, GDN08*, Coimbra, Portugal, June 2008.

[Knoll *et al.*, 2009] Stefan W. Knoll, Martin Hörning and Graham Horton. Applying a ThinkLet- and ThinXel-Based Group Process Modeling Language: A Prototype of a Universal Group Support System. In *Proceedings of the 38th Annual Hawaii International Conference on System Science, HICCS-42*, Big Island, Hawaii, January 2009. IEEE Computer Society.

[Nunamaker *et al.*, 1991] Jay F. Nunamaker Jr., Alan R. Dennis, Joseph S. Valacich, Douglas Vogel and Joey F. George. Electronic Meeting Systems to Support Group Work. *Communications of the ACM*, 34(7):40-61,1991.

[Nunamaker *et al.*, 1997] Jay F. Nunamaker Jr., Robert O. Briggs, Daniel D. Mittleman, Douglas R. Vogel and Pierre A. Balthazard. Lessons from a Dozen Years of Group Support Systems Research: A Discussion of Lab and Field Findings. *Journal of Management Information Systems*, 13(3):163-207, Winter 1997.

Towards Intelligent Adaptative E-Learning Systems – Machine Learning for Learner Activity Classification

Mirjam Köck
Johannes Kepler University Linz
A-4040, Linz, Austria
koeck@fim.uni-linz.ac.at

Abstract

As adaptivity in e-learning systems has become popular during the past years, new challenges and potentials have emerged in the field of adaptive systems. Adaptation, traditionally focused on the personalization of content, is now also required for learner communication and cooperation. With the increasing complexity of adaptation tasks, the need for automated processing of usage data, information extraction and pattern detection grows. We present learner activity mining and classification as a basis for adaptation in educational systems and discuss intelligent techniques in this context. Based on real usage data, we present the results of experiments comparing the behaviour and performance of different classification algorithms.

1 Introduction

This paper discusses how the goal of intelligent adaptive e-learning systems can be approached with the help of learner activity classification. Intelligent adaptive e-learning combines topics, issues and characteristics of various fields: e-learning, adaptivity and Computational Intelligence (CI).

E-learning environments have become fairly popular, as learning scenarios have been radically developing towards e-learning and blended learning during the past years. Almost every educational institution applies e-learning to a certain extent. Related systems usually include different kinds of facilities: tools accompanying learning content like exercises or assignments; and, communication tools like chat, fora or private messaging.

Adaptive systems offer various kinds of adaptation and personalization, most of which restricted to content (e.g. personalization of learning paths, recommendations of topics, and in some cases also a personalized view of the content). Mostly, adaptive e-learning systems have rather limited support for communication facilities and do not extend adaptation efforts to them. Recent attention to adaptive support for collaboration (see e.g. [Soller, 2007]) has been concentrating on research systems and has not been fully exposed to a large community as of yet.

Intelligent systems have been in the focus of attention for a longer period of time. Research combines principles like evolution, learning, in some sense also adaptation, fuzzy logic, etc. Intelligent systems are designed to simulate human reasoning and learning, reducing the need for human intervention in the application process. CI is promising for the further evolution of adaptive systems, especially in the context of e-learning where different learning theories [Lefrancois, 2006], [Prince and Felder, 2006], learning styles [Felder and Brent, 2005], and social processes [Vathanophas *et al.*, 2008] need to be addressed. As pointed out in [Brusilovsky and Peylo, 2003], educational systems are traditionally either intelligent or adaptive, listing prominent systems (like AHA! [De Bra and Calvi, 1998]) as adaptive but non-intelligent, and other ones as intelligent but limited regarding adaptivity.

In this paper, we focus on intelligent adaptive e-learning systems. Our approach relies on mining and processing of usage data. Usually, although activity data is monitored by the system, high levels of human intervention are required to process and use such data to achieve high-quality adaptation. We introduce an approach that is based on intelligent techniques for the classification of user activity data in e-learning environments and aims to largely supplement or even replace human efforts in this context.

The rest of this paper is structured as follows. Section 2 describes the state of the art and lists common problems in prevailing adaptive e-learning systems. Section 3 explains our classification strategy and how it can address the aforementioned issues. Section 4 compares statistical and CI-based approaches in the context of classification. Section 5 describes related experiments that were run to measure the performance of intelligent classifiers on learner activity data tasks. We summarize related work and give an outlook on future work in Sections 6 and 7.

2 The Adaptive E-Learning System - Two Pieces or One Whole?

Most e-learning systems consist of various kinds of tools which can be roughly categorized as learning facilities and facilities supporting communication/cooperation processes. In non-adaptive systems, tools are naturally independent. In adaptive ones, tools may require communication with other tools and/or a central service (e.g., an adaptation engine). In theory, this would be the basis for an integrated environment using knowledge gained in any of its facilities for system-wide adaptations.

Nevertheless, adaptive systems in the field of e-learning have been concentrating until now on some specific kinds of adaptation. In general, we can distinguish between adaptive navigation and adaptive presentation support [Brusilovsky, 1996]. These techniques are based on a user's interaction history within the system or additional information provided explicitly. They are well established, but, when it comes to e-learning, they have been primarily used to adapt content only.

Adaptation is often based on knowledge a system obtained from a user's interaction history, and that is then utilized to predict future activities which in turn become the basis for recommendations. At the moment, this information is typically not shared between different components of a system. For instance, a user's previous behaviour in the content facilities of a platform is only used to further adapt the content to the user's needs but not considered for guidance in communication tools. Therefore, adaptive e-learning systems are often not perceived as fully integrated, but rather an assembly of two independent pieces of a puzzle. A new approach would be to establish a shared pool of adaptation knowledge which is contributed to, and queried by all of a system's components.

3 Learner Activity Classification

Our general idea aims at introducing new kinds of adaptation in e-learning systems, bridging the common gap between content and communication facilities. Here, we approach this aim using activity mining and classification.

3.1 Activity Data

If we want to offer recommendations related to communication and learning content, we need to infer a user's level of interest in specific topics. Therefore, we examine a user's history on the system and use previous interests to predict future ones. We can shortly outline the concept as follows. First, we collect a user's passive ("consumption") activities. We will further also refer to this kind of activities as "read activities". Reading an element (e.g. an entry in a forum or a document) denotes a user's interest. If a user was interested in one specific element, we can find similar ones and assume that these are also interesting for this user. Given this kind of "knowledge", we can try to infer user interest for as many events as possible which can then become the basis for adaptation. This general idea can be put into practice by several different implementation approaches (see also Section 4) which provide different quality and granularity of results. All of them have in common that the primary objective is to classify data continuously produced by users' activities on a platform.

3.2 Classification Levels

We distinguish between two different levels of classification: classification of individual user activities, and classification of user activities considered as an interrelated construct. The first kind, as opposed to the second one, treats activities as if they were independent. The second kind is promising for modeling dependencies between users, tools, etc. but it requires a higher amount of reference information. We concentrate on the first kind here, which can partially be done before the system has collected enough information to generate reference constructs. It does not consider the time context of, and relations between, activities but uses activity items as independent of each other. Nevertheless, in most cases (depending on the learning technique) the system must still be provided a certain amount of reference data before classification of fresh data can be performed. This means that, in this case, no long period of training is necessary as long as some representative data sets are available. Therefore, the only prerequisite for this kind of classification is a certain period of data collection (depending on users' level of activity). Classification of independent activity items can be useful at the level of both individual users and groups (see also Section 7).

3.3 Application in Adaptive E-Learning Systems

First of all, we want to provide adaptation which closes the gap between learning facilities and those supporting communication and collaboration. This can be done by extracting information of all facilities, feeding it to one shared model which is then again queried by all facilities. Regarding the first level of classifictaion, we aim at recommending both communication threads and learning content items, based on a user's previous interests. For the second level of classification, our main application idea is closely related to group work. We want to be able to determine users' collaboration behaviour and their roles in group structures in order to recommend group constellations the system predicts successful on the one hand and interesting communication partners for individual users on the other.

4 Statistical vs. Intelligent Approaches

In order to classify independent user activity items we have to find an approach that is capable of computing realistic values for every user's interest in an event. There are several ways of approaching this, basically statistical and "intelligent" ones. The main characteristic of intelligence in this context is that the respective approaches are capable of learning, which is not possible for statistical ones. The statistical approach will work for some scenarios (in [Jung *et al.*, 2005], the authors introduce a statistical model for user preferences which performs well) but it can turn out to be too inflexible in others.

The aim is to not only determine interest, but also, on a higher level, provide recommendations of specific communication threads, learning material, etc. Knowledge about users gained in any of the platform's areas (communication or learning content) should be combined for the computation of interest levels. And finally, the system should of course continuously adapt to users' behaviour, i.e. all new actions must be considered. The following sections provide an introduction to each of the two approaches.

4.1 A Statistical Approach

This approach uses a statistical formula to compute a user's interest level for an item. The formula considers past user interest (indicated e.g. by read activities) to compute statistics which then becomes the basis for further prediction. First, the distribution of a user's read activities among tools in a site has to be computed. Basically, standard statistical metrics like mean, standard deviation, and variance are used to determine probability/density distributions. Given only the mean, we would face the problem of statistical outliers distorting the overall picture. This can be partially solved by considering the standard deviation (or variance). Given standard deviation, a tool's deviation σ_{T_x} from this value can be used to identify significant (in both directions) tool results. Consider the following simple example using 5 hypothetical tools and 25 read activities produced by 1 user within our time frame, distributed among the tools as $c_1 = 10$, $c_2 = 2$, $c_3 = 4$, $c_4 = 3$, $c_5 = 6$. Consider, we want to compute this user's interest value for every tool. This would result in the following:

$$\bar{x} = 5$$

$$\tilde{x} = 4$$

$$\sigma^2 = \frac{1}{5} * \sum_{i=1}^{5} (x_i - \bar{x})^2 = 8$$

$$\sigma \approx 2,83$$

In a next step, a tool-specific metric can be determined as

$$\sigma_{T_x} = |c_x - \bar{x}| - \sigma$$

which will mark all resulting $\sigma_{T_x} > 0$ as significant (in both directions). In our example σ_{T_1} and σ_{T_2} will be positive values, marking T_1 as significantly high (as $c_1 > \bar{x}$) and T_2 as significantly low (as $c_2 < \bar{x}$) regarding interest. This (simplified) approach can be improved, e.g. by adding weights, and in theory this improved version might be sufficient, but it still carries some non-obvious risks. For instance, a statistical formula, even if it contains variable elements, is inflexible, meaning that the core does not change for different scenarios. We have to be aware that users may behave differently in their interest across sites, tools or resources. There can be courses where communication plays a more important role than educational content, and users might differ in their communication and learning behaviour in several ways. Furthermore, we may want to weigh read activities differently based on the time when they occurred (e.g. if the timespan between the related ”active” create event and the read activity is relevant).

In order to consider all of these factors, the formula might have to look different for different combinations of users, tools, courses and resources. This is hardly possible, and even if it was, it would still lack the ability to continuously and individually adapt to a user's behaviour.

4.2 A Flexible, Self-Learning Approach

In order to overcome issues and problems raised by purely statistical approaches, classification techniques of the field of machine learning can be used. These techniques do not make as many semantic assumptions as statistical approaches do, but learn from the user. Although the classifiers we used for our experiments (see a detailed description in Section 5) differ drastically in their way of model building, they have in common that their models consider all features we provide as input. In our case, 8 attributes are available, 6 of which (the anonymized user id, event id, tool id, site id, related resource and the interest class) are taken into account by the classifiers. The remaining two, index and timestamp, were removed by a filter in preprocessing because we do not consider temporal relations for this kind of classification yet. This means, all solutions we get dynamically adapt to all feature values of new input events. Thus, not only the site where the event occurred is considered, but also e.g. its creator and the tool where it originated. To further extend flexibility and personalization, the classifier then computes an event's interest value for every user individually. This implies that the approach works separately for every user. As the classifier is continuously fed with new information, it is able to learn and adapt its behaviour during the process. As each of the classifiers builds a model (e.g. a decision tree, a Bayesian Network, or a rule base) which can be queried, it is also possible to extract semantic information from it which will offer additional knowledge about users, behaviours, and the whole construct of content, courses and tools. In addition, dependencies and correlations between attributes could be found which might become important for further event design. Especially the opportunity to gain semantic information from the model built by a classifier is a significant advantage compared to a statistical approach, as the latter is limited to strict one-way information exchange, i.e. no information can be extracted from statistics in a way it can go back into and enhance the user model.

5 Experiments

This section describes experiments designed to test our classification approach on real user activity data, compare the performance of different techniques for different aims, and show how classification can improve activity-based adaptation. The experiments aim at producing a group-based interest model. In general, we can distinguish between user- and group models. A user model is created for every user individually and only fed with information about that specific user. A group model pictures group behaviour, i.e. activities of multiple users which were clustered into groups (e.g. , based on similarities, or a given course context). Our system is fed with all users' activities and tries to classify new events as interesting or non-interesting for every user individually, but uses this knowledge to build a shared model. This model will be referred to in later stages of our work to offer group-based adaptations. We can benefit from working with group models in several ways. For example, to avoid the ”cold start” problem [Höök, 1997], a group model can become the default for a new course participant. The system then does not have to create new models from scratch any more but can build upon one based on the interest and activities of a group working on the same content and tasks.

5.1 Setup

Our experiments outline an extension to the behaviour of the ”recent activity tool” [1] which is an add-on to the e-learning platform Sakai [Sakai, 2009]. This tool provides an overview of recent activities in various Sakai tools. It includes a personalized view marking activities as interesting for the current user, and a personalized RSS feed. The tool adapts at the user-level only at the moment. Adaptation is not done before the system has received a sufficient amount of information about the user. Recommendations are based on a statistical model similar to the one described before. Thus, the adaptive part of the recent activity tool relies on assumptions and generalizations to a certain extent. As already described, CI techniques can improve the performance, flexibility, and accuracy of adaptive components because they learn from the user instead. Our experiments use these techniques to replace the statistical model. Real usage data is provided by a monitoring extension to Sakai. The instances are independent and handled as random set elements for the first run of experiments. Yet, they contain information which can help to create relations in further post-processing.

The overall data set contains 4967 instances with 6 features as described in Section 4.2. String attributes were normalized to nominal ones, meaning that before data went into the classifiers, a filter collected all possible values for a feature (for instance, all tools where activity was monitored). The resulting finite set of values then allows for better computation of probabilities.

The experiments were run on data of one specific course about the Unified Modeling Language, with 31 participants in total. Data was collected over a period of several months and went through some preprocessing during which irrelevant or pseudo-data (e.g. produced by test users) was removed. During these steps anonymization was also performed by encoding user IDs with a one-way hashing algorithm.

[1] The recent activity tool was developed in the context of the Adaptive Learning Spaces (ALS) project. For further information, please refer to http://www.als-project.org

5.2 Process and Technologies

Experiments were carried out iteratively with training, testing and evaluation steps repeated for different classification algorithms. Validation was performed in two different ways – by 10-fold cross-validation, and by specifically split training- and test sets. A comparison of the algorithms' results concludes the experiments and becomes the basis for classifier rating and final selection. We used the Weka [Witten and Eibe, 2005] machine learning software to run the experiments. For a more detailed description of the algorithms please refer to Weka documentation and tutorials. The following paragraphs describe the configuration of the classification algorithms which were used.

Naïve Bayes: The naïve Bayesian approach builds a simple network with one parent node (the class label, in our case the interest value). There are no important additional configuration alternatives.

Bayesian Network: The network applied for the experiments uses the SimpleEstimator approach for finding the conditional probability tables of the net. The TAN algorithm (determining the maximum weight spanning tree and returning a Bayesian Network augmented with a tree) is applied for searching network structures.

SMO (Sequential Minimal Optimization): SMO is used to train a support vector machine. We used standard settings with relatively low complexity (the higher, the fewer wrong classifications are accepted) and a polynomial kernel $K(x, y) =< x, y >^p$ with exponent $p = 2$.

Multilayer Perceptron (Backpropagation Neural Network, later referred to as *NN*)**:** We used a network with $a = attributes + classes$ hidden layers of sigmoid nodes, a learning rate of 0.7, momentum of 0.2 and 500 learning cycles. Please note that run-time filters like nominal to binary slow down the process significantly.

IBk (Nearest Neighbour): We used $k = 10$ and the LinearNearestNeighbourSearch (brute force) algorithm for nearest neighbour search and cross-validation.

JRip (Rule-based): This algorithm implements a propositional rule learner and provides a set of rules which are then used as a basis for classification decisions. Our experiments use 10 folds (for pruning and growing rules) and 6 optimization runs.

J48 (Tree-based): This algorithm, building a decision tree, uses a confidence factor (small values mean more pruning) of 0.25 and reduced error pruning here.

RandomTree: This algorithm, building a decision tree, uses a KValue (i.e. the number of randomly chosen attributes) of 1 and an unlimited tree depth.

5.3 Results

Table 1: This table lists classification results of various algorithms on the overall data set (10-fold cross-validation). The NN classifier is listed twice, once with filters. The table further displays the percentage of correctly classified positive instances, the True Positive rate for class 1, the Root Mean Squared Error, the time taken to build the model T_m, and the time taken for the overall process T_o.

Class.	***Corr.***	***TP***	***RMSE***	T_m ***(s)***	T_o ***(h,m,s)***
NB	47.6%	76.1%	0.1550	$< 0.01s$	$< 1s$
BN	70.0%	72.3%	0.1112	$< 0.01s$	$< 1s$
SMO	70.4%	84.5%	0.1261	$105.39s$	$16m54s$
NN(f)	70.3%	12.3%	0.1668	$17.22s$	$21h16m8s$
NN	56.0%	60.6%	0.1366	$17.16s$	$2m51s$
IBk	70.4%	84.5%	0.1092	$< 0.01s$	$5s$
JRip	68.0%	85.2%	0.1143	$0.63s$	$7s$
J48	70.7%	74.8%	0.1137	$< 0.01s$	$< 1s$
RT	70.4%	84.5%	0.1092	$0.13s$	$< 1s$

The results of the described base experiment are listed in Table 1 and Figure 1. The base experiment uses 10-fold cross-validation to get a first impression of the classifiers' performance. Subsequently, more specific experiments were conducted in order to find out how their performance changes over time. The experiments were conducted on a 2,98 GHz dual-core machine with 4 GB RAM, running 64-bit Windows XP. As a first experimental step, we compared the performance of different classification techniques to the performance of a statistical approach as described in Section 4.1. The percentage of correctly classified instances ranges from 96.63 (Naïve Bayes) to 98.41 (SMO) for the machine learning techniques. The statistical model obtains a result of 68.94%. In the following, we do a more detailed comparison of the classifiers listed in Section 5.2. The percentage of correctly classified instances from now on refers to "positive" instances (i.e. the instances with an interest value of 1) only. The overall results, containing "negative" instances also are less expressive, as the number of these instances is higher and their classification much easier (for the CI techniques only). This leads to a very similar overall performance of the classifiers and subsequently to a misleading picture and potentially wrong conclusions. The results show that the classification task itself can be handled relatively well by different classification techniques. As there is only little discrepancy regarding the number of correctly classified instances, process time becomes an even more important criterion. After running experiments with cross-validation, which provided a first general impression on the performance of machine learning techniques on our data, we modeled a similar experimental setup in order to measure how fast the classifiers learn from given input data. The experiments were run on split data (training set and test set). As we are building a group model, the percentage split

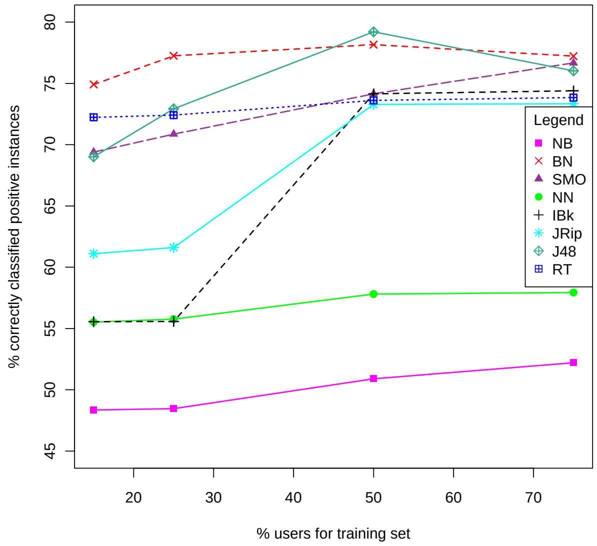


Figure 1: This plots show how the performance of classifiers increases with an increasing amount of training data.

for the data set is based on users, not resources. This means that the training set does not contain a certain percentage of the data but all data of a certain percentage of users. We ran the experiments several times with the events for 15%, 25%, 50%, and 75% of the users as training and remainder as test set. As depicted in the plots (Figure 1), the results show three different trends. Bayesian Network, RandomTree and J48 (the last two both tree-based approaches) show good classification performance right from the beginning and relatively steady behaviour. SMO could also be added to this "cluster" of algorithms, regarding its effectiveness. Next, we can see a second cluster containing IBk and JRip. These algorithms show good results but not right from the beginning. However, their plateau is at about the same place as the first cluster. The third trend can be seen in MultilayerPerceptron (NN) and Naïve Bayes which are steady in their performance but don't provide promising results. This means that for subsequent work we will not consider the classifiers of the third cluster. IBk and JRip will be further explored, but the most likely candidates are those in the first cluster, where the favourites are Bayesian Network and the tree-based classifiers. SMO, compared to the other classifiers, is relatively slow, with the time needed to build a model increasing at least linearly as the training set grows. In general, a linear algorithm is reasonable for run-time employment. In our case, comparing SMO to the faster classifiers, the discrepancy in computation complexity (< 0.01 seconds as opposed to 1.15 seconds for building the model for the smallest training set) is significant enough to be an exclusion criterion. SMO will be kept for further observation, but does not remain a first choice candidate.

Another important criterion for the selection of a classifier is in our case the possibility of information extraction, given a model. Descriptive classifiers like Bayesian Networks, rule- or tree-based approaches enable very simple extraction of semantic information, whereas function-based ones like neural networks or support vector machines tend to behave like blackboxes. Generally we can conclude that learning classifiers perform well on our data. Therefore, also considering the issues and potential problems and limitations of statistical approaches (see Section 4), these techniques are highly promising for our scenario and all subsequent ones operating on data of a similar structure.

6 Related Work

Our general approach is based on a combination of the fields of adaptive systems, e-learning and CI. Thus, we do not only have to consider challenges of the particular areas but also the potentials lying in the aggregation. This is not the first attempt pointing in that direction. For instance, our work relates to recent research issues in the field of adaptive collaboration support as described in [Paramythis, 2008]. In general, the matter of distributed collaboration entails some challenges. Their specific effects on the development process regarding adaptive support was e.g. elaborated in [Soller, 2007] where the author also describes relevant social processes. Additionally, personalization in distributed environments is further discussed in [Dolog *et al.*, 2004], where the authors introduce recent projects and also address personalization on the Semantic Web.

Regarding Machine Learning (ML), we can refer to research described in [Webb *et al.*, 2001], where the authors particularly treat student modeling and explicate specific requirements of this area. A concise overview on data mining techniques from the perspective of adaptive systems is given in [Voges and Pope, 2000].

Regarding the context of data mining in education, we find particularly interesting results in [Romero *et al.*, 2008], where the authors compare different algorithms to classify students. They also describe experiments aiming at predicting students' final grades based on usage data. The selected set of algorithms is partly congruent to ours, but operates at the user level instead of the activity level as in our approach, i.e. their data set contains items already aggregating information about user activities. This approach seems perfectly sound at the first glance, but it is less flexible as only a specific number of information elements can be considered which makes it hard to add further semantics later if necessary. As both approaches use semantically similar data but for different objectives and on a different level, it is very interesting to compare the results. Some trends can be found in both reports, whereas in other areas there is relatively high discrepancy. For example, the authors argue against e.g. Neural Network and Nearest Neighbour classifiers in their scenario in particular and data mining in general, due to the lack of comprehensibility. As these classifiers are in cluster two and three in our evaluation, we agree with them here, although the Neural Network achieved a better classification result on their data. Our second classifier in cluster three, Naïve Bayes was not included in their study. Tree-based classifiers performed very well in both cases. Unfortunately, the performance of Bayesian Networks cannot be compared because it was not included in the evaluation of Romero et al. However, they did an additional step of comparing the classifiers' performances on "plain" data to those after preprocessing and identified what classifiers can actually be improved by preprocessing, which we will consider during our next steps.

Further related work can be found in [Oakley *et al.*, 2004], where the author describes data-driven modeling of students' interactions, aiming at predicting students' ability to correctly answer a question and whether a student's interaction is beneficial in terms of learning. The experiments focus on Bayesian Network models. Additionally, we can also find interesting information in research on statistical approaches in machine learning, which is relevant for our approach because it identifies scenarios where statistical approaches work particularly well. Find a description of a statistical rule learning approach in [Rückert and Kramer, 2006]. In [Jung *et al.*, 2005], a detailed comparison of statistical and non-statistical approaches is given.

7 Conclusions and Future Work

As the experiments have shown, flexible classification approaches perform well on user activity data as produced on a learning platform like Sakai. The results can potentially be improved by combining (complementary) classifiers (using ensemble methods like bagging, boosting or stacking). The solution is not restricted to Sakai or even the recent activity tool, as data of any learning environment can easily be converted to a similar format. The intelligent classification approach is extendable in several ways. First, it can be applied on different levels, building models for individual users, groups or other clusters (e.g. any specifically interesting combination of features). Second, as described in Section 3, classification is not restricted to individually handled events, it can also be applied at the level of activity paths. These paths, representing a sequence of (related) instances, are a way of modeling relations between activities or any of their features. As a next step we will concentrate on modeling users and their collaboration behaviour with this approach. Several issues have to be considered:

Some factors in the path building process are strongly dependent on specific features. For instance, the timespan between the occurrence of subsequent events must be handled differently for various tools. In a synchronous communication tool, like a chat, an event which occurs hours after another one is more likely to be independent than in an asynchronous communication tool, like a forum, where the context is more important that time. In order to avoid wrong conclusions due to similar conditions, we have to set up a knowledge base containing factors and their respective values which may vary for different tools, etc.

What are the concrete questions we want to be able to answer given the path model? Before any design-specific decisions can be made, we have to define what should be modeled, like e.g. the level of communication between users or the context-based relations between communication and learning content tools.

What kind of data representation is best suitable for the model? Given information about semantics of the model and requirements for the information which should be extracted from it, adequate representation must be chosen. There are several ways of modeling entities, relations and weights, such as graphs. Implementing and evaluating an approach based on a combination of matrices, graphs and a set of new metrics (measuring e.g. the degree of so-called parental relationships between users or other features) will be the next step in the process. The aims include modeling collaboration, defining metrics indicating ”success”, classifying the outcome as successful or not and, during the process, finding out what leads to successful collaboration and what has adverse effects.

In synthesis, we can state that our CI-based classification techniques are promising in several ways. Not only are they capable of replacing strongly assumption-based approaches and thus improve the performance and flexibility of adaptive features; in addition, we can potentially overcome the problem of a gap between different kinds of facilities in adaptive e-learning systems as introduced in the first sections. We consider data produced in practically any different kinds of tools, and, once the model is integrated in the learning environment, it can also be queried from all the system's components. Thus, knowledge about a student gained in one area can be used for adaptations in others. Moreover, using our extended classification approach operating on interrelated data, we can easily model relations between tools and other features which can become the basis for new kinds of adaptation and recommendations.

Acknowledgements

The work reported in this paper is funded by the ”Adaptive Support for Collaborative E-Learning” (ASCOLLA) project, supported by the Austrian Science Fund (FWF; project number P20260-N15).

References

[Brusilovsky and Peylo, 2003] P. Brusilovsky and C. Peylo. Adaptive and Intelligent Web-Based Educational Systems. *Int. Journal of Artificial Intelligence in Education*, 2003.

[Brusilovsky, 1996] P. Brusilovsky. Methods and Techniques of Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 6(2–3):87–129, 1996.

[De Bra and Calvi, 1998] Paul De Bra and Licia Calvi. AHA! An open Adaptive Hypermedia Architecture. *The New Review of Hypermedia and Multimedia*, 4, 1998.

[Dolog *et al.*, 2004] Peter Dolog, Nicola Henze, Wolfgang Nejdl, and Michael Sintek. Personalization in Distributed E-Learning Environments. In *WWW (Alternate Track Papers & Posters)*, pages 170–179, 2004.

[Felder and Brent, 2005] Richard M. Felder and Rebecca Brent. Understanding Student Differences. *Journal of Engineering Education*, 2005.

[Höök, 1997] Kristina Höök. Evaluating the Utitlity and Usability of an Adaptive Hypermedia System. In *Proceedings of the 1997 International Conference on Intelligent User Interfaces*, pages 179–186, 1997.

[Jung *et al.*, 2005] Sung Young Jung, Jeong-Hee Hong, and Taek-Soo Kim. A Statistical Model for User Preference. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 2005.

[Lefrancois, 2006] Guy R. Lefrancois. *Psychologie des Lernens*, volume 4. Springer, Heidelberg, 2006.

[Oakley *et al.*, 2004] Barbara Oakley, Rebecca Brent, Richard M. Felder, and Imad Elhajj. Turning Student Groups into Effective Teams. *Journal of Student Centered Learning*, 2(1), 2004.

[Paramythis, 2008] Alexandros Paramythis. Adaptive Support for Collaborative Learning with IMS Learning Design: Are We There Yet. In *Proceedings of the Workshop on Adaptive Collaboration Support*, 2008.

[Prince and Felder, 2006] Michael J. Prince and Richard M. Felder. Inductive Teaching and Learning Methods: Definitions, Comparisons, and Research Bases. *Journal of Engineering Education*, 2006.

[Romero *et al.*, 2008] Cristóbal Romero, Sebastián Ventura, Pedro G. Espejo, and César Hervás. Data Mining Algorithms to Classify Students. In *Proceedings of the 1st International Conference on Educational Data Mining (EDM08)*, 2008.

[Rückert and Kramer, 2006] Ulrich Rückert and Stefan Kramer. A Statistical Approach to Rule Learning. In *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, 2006.

[Sakai, 2009] Sakai. http://www.sakaiproject.org, 2009.

[Soller, 2007] A. Soller. Adaptive Support for Distributed Collaboration. In *The Adaptive Web*. 2007.

[Vathanophas *et al.*, 2008] Vichita Vathanophas, Kingkarn Suensilpong, and Tullawat Pacharapha. Task-Related Information Sharing in Group Support Systems (GSS). In *Proceedings of the European and Mediterranean Conference on Information Systems 2008*, 2008.

[Voges and Pope, 2000] K.E. Voges and N.K. Pope. An Overview of Data Mining Techniques from an Adaptive Systems Perspective. In *Proceesings of ANZMAC 2000: Australian and New Zealand Marketing Academy Conference*, Gold Coast, Australia, November 2000.

[Webb *et al.*, 2001] G.I. Webb, M.J. Pazzani, and D. Billsus. Machine Learning for User Modeling. *User Modeling and User-Adapted Interaction*, 2001.

[Witten and Eibe, 2005] Ian H. Witten and Frank Eibe. *Data Mining*. Morgan Kaufmann, 2005.

New Tagging Paradigms for Enhancing Collaboration in Web 2.0 Communities

Andreas Nauerz
Matthias Brück
Martin Welsch
IBM Research and Development
71032 Böblingen, Germany
{andreas.nauerz|mbrueck|martin.welsch}
@de.ibm.com mail

Fedor Bakalov
Birgitta König-Ries
University Jena
07743 Jena, Germany
{fedor.bakalov|koenig}
@informatik.uni-jena.de

Abstract

In this paper we present new sophisticated tagging paradigms and their influence on users collaboration behavior and the construction of user– and context–models.

We present paradigms like alien tagging which allows one user to apply tags for another user, reputation-based tagging which allows users' expertise to influence tags' weights, quantitative tagging which allows users to manually manipulate tags' weights, anti tagging which allows users to specify "negative tags", tag voting to solve the tag space littering problem by e.g. allowing users to vote against tags, tag expiry which allows tags to have a lifetime, contextual tagging which allows tags to be associated to certain context profiles, and so forth and describe how these can be used to refine our models and to perform even more valuable adaptations or to issue more valuable. We also allow for mechanisms to follow users' tagging "trails" in order to learn from what they are tagging.

All these techniques aim to provide the user with more advanced ways, to add, filter, group and view tags.

The concepts presented are currently been prototypically implemented within IBMs WebSphere Portal and can be presented in a live demo at the workshop.

1 Introduction

In recent years Enterprise Information Portals have gained importance in many companies. As a single point of access they integrate various applications and processes into one homogeneous user interface. Today, typical Portals are comprised of a huge amount of content. They are no longer exclusively maintained by an IT department, instead, Web 2.0 techniques are used increasingly, allowing user generated content to be added. These systems grow quickly and in a more uncoordinated way as different users possess different knowledge and expertise and obey to different mental models. The continuous growth makes access to really relevant information difficult. Users need to find task- and role-specific information quickly, but face information overload and often feel *lost in hyperspace*. Thus, users often miss out on resources that are potentially relevant to their tasks, simply because they never come across them. On the one hand, users obtain too much information that is not relevant to their current task, on the other hand, it becomes cumbersome to find the right information and they do not obtain all the information that would be relevant.

The recent popularity of collaboration techniques on the Internet, particularly tagging and rating, provides new means for both semantically describing Portal content as well as for reasoning about users' interests, preferences and contexts. It can add valuable meta information and even lightweight semantics to web resources.

In our previous work [Nauerz *et al.*, 2008] we proposed a framework which allowed arbitrary annotators, e.g. human users or analysis components (for automated tagging), to annotate any of these resources. Analysis of the tagging behavior allowed us to model interests and preferences of users as well as semantic relations between resources, and thus to perform reasonable recommendations and adaptations.

In this paper we present paradigms like alien tagging which allows one user to apply tags for another user, reputation-based tagging allows users expertise to influence tags' weights, quantitative tagging which allows users to manually manipulate tag's weights, anti tagging which allows users to specify "negative tags", tag voting to solve the tag space littering problem by e.g. allowing users to vote against a tag, tag expiry which allows tags to have a lifeime, contextual tagging which allows tags to be associated to certain context profiles, and so forth and describe how these can be used to refine our models and to perform even more valuable adaptations or to issue more valuable. We also allow for mechanisms to follow users tagging "trails" in order to learn from what he is tagging.

2 Related Work

A lot of work is currently underway to experiment with different techniques to improve working with tag engines.

Most researchers try to improve the quality of tags being presented. Tagging systems must often select a subset of tags to be displayed to the user due to limited screen space. Thus they must determine the most valuable ones. [Sen *et al.*, 2009a] present a tag selection algorithm based on users' implicit and explicit (tag rating) behaviour, to select the right tags to be displayed. [Liu *et al.*, 2009] present a tag ranking algorithm used on Flickr[1]. Other approaches for tag selection algorithms are presented in [Sen *et al.*, 2009b] and [Zhang *et al.*, 2009]. Other researchers aim to improve tag quality by recommending and suggesting

[1] http://www.flickr.com

tags. Such approaches are e.g. described in [Suchanek *et al.*, 2008]. [Garg and Weber, 2008a] present a new algorithm called *Hybrid* to determine reasonable tags to be recommended to users on Flickr. [Symeonidis *et al.*, 2008] present a tag recommendation algorithm based on tensor dimensionality reduction. Other tag recommendation work is described in [Garg and Weber, 2008b], [Sigurbjörnsson and van Zwol, 2008], [Song *et al.*, 2008], and [Vig *et al.*, 2009].

Other work goes a little bit more into the direction of what we do and provides users with new means to directly influence tag quality. [Lee and Han, 2007] introduce *QTag* a qualitative tagging system that allows users to tag in order to rate content and express opinions.

Other work similar to ours also experiments with new visualization techniques. [Gwizdka and Bakelaar, 2009] presents a technique for preserving and presenting context and history while navigating web resources described by keywords using tags and tag clouds as application area.

Many researchers also try to improve tag quality by enriching tags with more semantics. [Echarte *et al.*, 2009] introduced methods to group tag variations with matching techniques. Another collaborative Web 3.0 approach was presented by Kreiser et al., which allows users to augment tags with semantics and collaboratively model relations between tags.

In the end quality tags are often used not only to recommend other tags but finally to recommend content. This kind of personalized recommendation of content is based on the content's relatedness to certain tag terms. E.g., [Wu *et al.*, 2006] proposes a modified version of the HITS algorithm to determine experts and high-quality documents related to a given tag.

3 Concepts

3.1 Alien tagging

As said before Web 2.0 communities can be rather heterogeneous. The expertise of users contributing (and consuming) content can vary a lot. What might be obvious for one user might be completely unknown to others. *Alien tagging* allows more experienced users to tag content for less experienced ones. In our prototypical implementation tag widgets allow power users to apply tags to resources on behalf of other users (or even user groups). Next time one of the users for which alien tags have been applied logs-in, he or she is notified about the availability of these and can inspect the underlying resources. The same way we used "normal" tags in our previous work [Nauerz *et al.*, 2008] to refine user models that describe users interests and preferences we can use these alien tags, too. In real environments *alien tagging* could be used e.g. by managers pretagging content for their new hires, by team- or technical leads to point their team members to relevant content which they otherwise might have missed. Thus *alien tagging* opens another opportunity to prevent users from missing out content by issuing recommendations provided by "alien" users. To identify the different kinds of tags in the tag cloud, each kind is encoded in a different color (figure 1). Green tags are tags added and only visible to the user who applied them, the private tags. Blue tags are tags added and visible to the whole community, the public tags. Orange tags are added from one user for another user, the alien tags. An alien tag has two additional icons in the upper right corner, a *plus* sign to transfer the alien tag to the private tag store of a user and the cross sign to discard the alien tag (afterwards, the tag is deleted from the tag cloud). An alien tag is applied by selecting the target user from a drop-down box and specifying the tag name (figure 2).

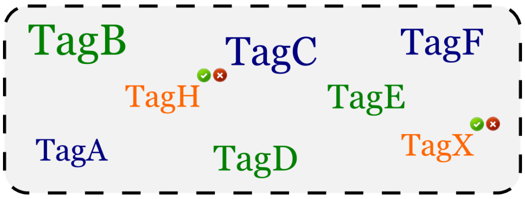
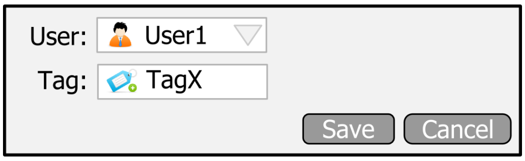


Figure 1: View alien tags in the tag cloud

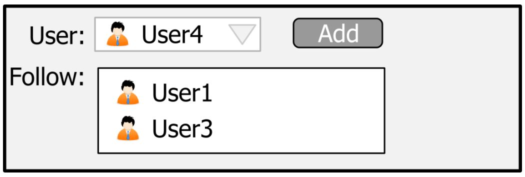

User: User1

Tag: TagX

Figure 2: Apply an alien tag

3.2 Tag following

As mentioned previously expertise and interests of users in a community can vary a lot. Therefore it can be interesting for a user to see how other taggers work with the tagging system and in what kind of resources they are interested. If a user thinks that he could benefit from following other taggers, because of overlapping interests with respect to some topics or just to see in what kind of topics a more experienced user is interested in, we provide him with means to follow the tagging "trail" of this user. The user just needs to select the user to follow from a drop-down box (figure 3). Afterwards a notification informing about any newly added public tag of the user being followed pops up, every time the user following logs in (figure 4). The following user can visit the resources and even transfer the tags of the user being followed to his own tag store. It is also possible to follow only certain kinds of tags, e.g. a specific tag bag (see section 3.6), to narrow the focus of the tagging stream.

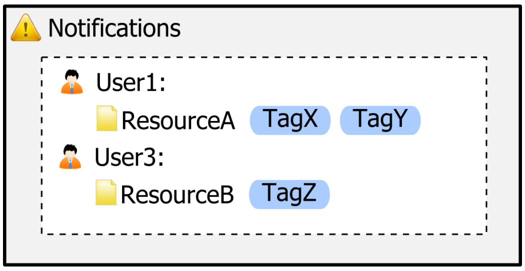

User: User4

Follow:
User1
User3

Figure 3: Follow a users tagging stream

3.3 Reputation-based tagging

In our previous solutions we always assumed that the weight (i.e. the importance) of tags only depends on the frequency of their occurrence. I.e. a tag applied more often with respect to a certain scope was regarded of higher importance than a tag applied less often. In our new prototype we additionally assume that the weight of a tag can depend on the reputation (or expertise) of a user. I.e. that tags applied by more experienced users have higher weights, and thus higher influence on what content the community is presented (or recommended) with, than tags from less


⚠ Notifications

User1:

ResourceA TagX TagY

User3:

ResourceB TagZ

Figure 4: Notification about newly added tags

experienced users. This way we can point users to more relevant content as we assume experts to know better what the community should focus on. E.g., in development team we assume the tagging behavior of the team- or technical lead of higher importance. With *reputation-based tagging* we also ensure that ”incorrect or less suited” tags perceive lower weights (influence). E.g., a newbie might apply a more ”incorrect/less suited” tag as he just misunderstands (due to his insufficient knowledge) what he is looking at. The way we determine users’ expertise has already been described in [Nauerz *et al.*, 2008]. In figure 5 the weight of the tags displayed in the tag cloud only reflect the count of the tag. The magenta colored tags are tags applied by user ”UserA” and the cyan colored tags are applied by user ”UserB”. The tag cloud in figure 7 also considers the reputation level of the user, which applied the tag, in order to calculate tag weights. Therefore we allow users to apply ratings to tags and to users of the community. The reputation level of a user can be determined by, e.g. calculating the median over all ratings applied to the user or over all ratings applied to tags this particular user has applied. Figure 6 shows, that user ”UserB” got a better average rating then user ”UserA”. We see the impact of this difference in both users reputation in the tag cloud in figure 7. Even though, the tag ”TagB” is applied as often as the tag ”TagF”, it is displayed with a lesser weight than tag ”TagF”, just because of the better reputation level of ”UserB” compared to ”UserA”.

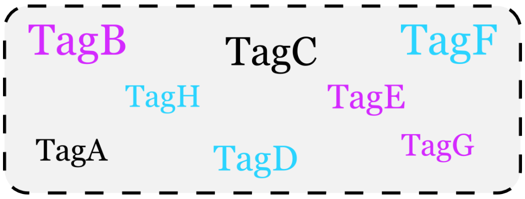


Figure 5: Tag cloud without reputation based weighting

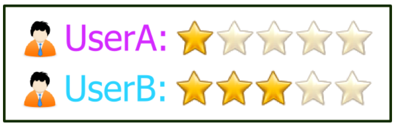


Figure 6: Average rating of user ”UserA” and ”UserB”

3.4 Quantitative tagging

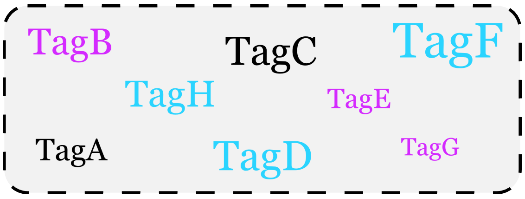


Figure 7: Tag cloud with reputation based weighting

Previously we also assumed that tags can only have ”positive character”. I.e. that we assumed that a resource can be tagged with a term to describe that the resource has something to do with this term, but also assumed that a resource cannot be tagged with a term to describe that the resource has nothing to do with it. In addition to that aspect we did not provide means allowing single users to express that a certain tag is of less relevancy for them. *Quantitative tagging* provides a solution to both problems: in our prototypical implementation a *plus*- and a *minus* sign is presented besides each tag being displayed. In addition, when applying a tag, a *not* sign is available (figure 9). Clicking the *not* sign when applying a tag allows users to express that a resource has nothing to do with the term applied, a helpful feature for more fine-granular categorization of resources: e.g., users could tag some resources with the term *Web 2.0* and a few of them with ”not” *scientific*. This helps users to quickly find all Web 2.0 related resources and to quickly distinguish between the scientific and non scientific ones among them. Clicking the *plus*- and *minus* signs when working with tags allows single users to express that they are less interested in a tag (or a certain tag associated to a certain resource) or can additionally express that a tag is of less relevancy for the entire community (figure 8). In the tag cloud an anti tag is displayed red colored and with a not sign in the upper right corner . Thus, these mechanisms allow for further refinements of our user models.

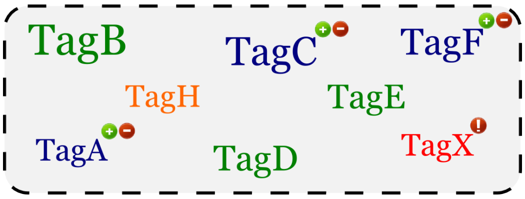


Figure 8: Rate a tag

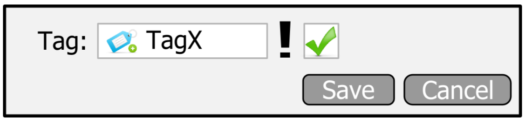


Figure 9: Create a ”Not-Tag”

Anti tagging

Anti tagging describes an enhancement to *quantitative tagging* (cp. 3.4). Here we automatically increase or decrease tags’ relevancy for the entire community by analyzing tags semantics (cp. [Nauerz *et al.*, 2008]). One option we have evaluated is to take into consideration antonyms. E.g., when a resource is tagged with ”good” and ”bad” we regard it as not tagged at all with either of these two terms as they annihilate each other. Antonyms can e.g. be found using the antonym thesaurus [2]. As *anti tagging* is not trivial to be realized as most examples are much more complicated

[2] http://www.synonym.com/synonyms/

and less obvious than the one just provided we have not yet incorporated it in our prototype.

Tag voting

Tag voting is a further enhancement to *quantitative tagging* (cp. 3.4). A user can vote in favor of a tag in order to let it become a favorite tag or against a tag. While the user hovers over a tag, a click on the heart icon (figure 10), expresses that he regards the tag as important and should thus be higher weighted than the other tags. The tag is automatically stored in a tag bag called *My Favorites* providing quick access to all favorite tags. A click on the trash icon (figure 10), indicates that the user regards the tag as inappropriate to describe the resource it has been assigned to. We provide another seperate view which is free of the tags the user has voted against. Depending on the system configuration if enough user vote against a tag the tag can be entirely removed from the resource. Thus, voting against tags gives the community the power to correct errors and solve the tag space littering problem autonomously.

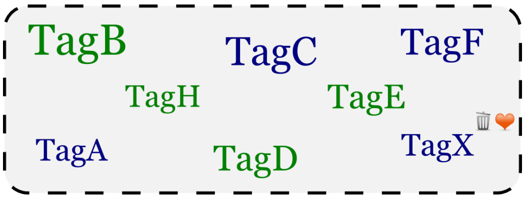


Figure 10: Vote against a tag or favor a tag

3.5 Tag expiry

In our previous work we also assumed that tags can be applied once and stay alive until they are manually deleted again. This let to tag-space littering as most users never deleted tags anymore even if they became obsolete. The fact that tags do not remain valid forever occurs in Portals that provide dynamic content very often. This resulted in having a lot of tags assigned to resources that did not describe the resource adequately nor express the resources relevancy to the community appropriately anymore. In our prototype *tag expiry* allows users to specify a chronological validity for tags when assigning them to a resource. Taggers can give tags a start date, an end date or a time frame in between they live. We also allow tags that are assigned a ”lifetime” to become more (or less) important as time passes by. E.g. if there is a page in the Portal system providing information about the Olympic Games 2012, this page might become more and more interesting to users as we get nearer to the year 2012 and less interesting after 2012. Thus users can specify that the tag should not be available before 2011, vanish after 2013 and become more important from 2011 till 2012 and less important from 2012 till 2013. Thus, *tag expiry* is yet another mechanism to help the community to focus on what is currently really relevant. Moreover, *tag expiry* allows us to neglect ”invalid” tags from being considered when doing content adaptation or recommendation. A clock icon in the upper right corner of a tag in the tag cloud indicates that a lifetime has been applied to the tag (figure 11). If only one user has applied a lifetime to the tag, a tooltip appears during hovering over the tag displaying the dates of the lifetime. Otherwise, the clock icon implies that multiple lifetimes from different users have been applied to the tag. The tag cloud can be filtered by specifying a date or dragging the date slider into the past or future. Latter feature allows simulating past or future representations of the tag cloud; i.e. it provides ”filtered” views of the cloud with respect to a certain point in time. To apply a tag with a lifetime, the user selects, using a date picker the start date, the end date or both (figure 12).

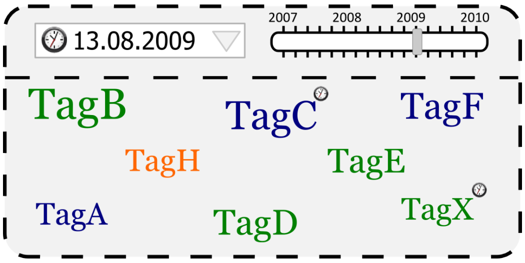
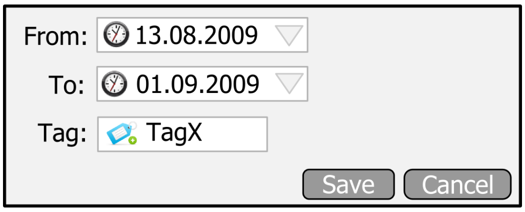


Figure 11: View tags with lifetime in the tag cloud


From: 13.08.2009

To: 01.09.2009

Tag: TagX

Figure 12: Apply a tag with specific lifetime

3.6 Tagging tags and meta-tagging

Previously we have also worked on solutions to solve major problems of tagging systems: most of these problems discussed dealt with synonyms (multiple tags having the same meaning) and polysemies (a single tag having different meanings). Current tag engines often try to overcome these issues by applying stemming and normalization algorithms which most often only solve problems resulting from morphological variations. Semantical variations can most often not be detected to be a synonym e.g. In our latest prototype we allow the community to resolve the resulting tag-space littering. In our tag-clouds we allow users to drag and drop tags on each other to consolidate them (figure 13). If a tag is dragged onto another tag, the user specify which one of the tags is the representative tag. In addition to that we allow users to create meta-tags (or meta-tag bags as we call them) under which other tags can be organized (figure 14) .Users can create private meta-tag bags only they can see or community meta-tag bags all users part of the community can see. That way users can e.g. create a meta-tag bag ”sports” and drag all sports related tags into that bag; users can also create a meta-tag bag ”favorite-stuff” and just drag what he/she likes most into it.

3.7 Contextual tagging

We also allow for contextual tagging where we can associate tags a certain context (for our context modeling approaches refer to [Nauerz *et al.*, 2008]) to prevent irrelevant tags (irrelevant in a certain context) to appear. This helps focusing on currently relevant content again.

Summarized, a context is described by a context profile which is defined by a set of context attributes permanently observed (e.g. location, day of week, end user device being used to access the system, and so forth). Context management portlets can be used to define new context profiles

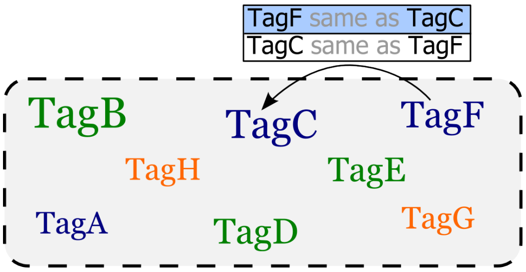


Figure 13: Consolidate two tags

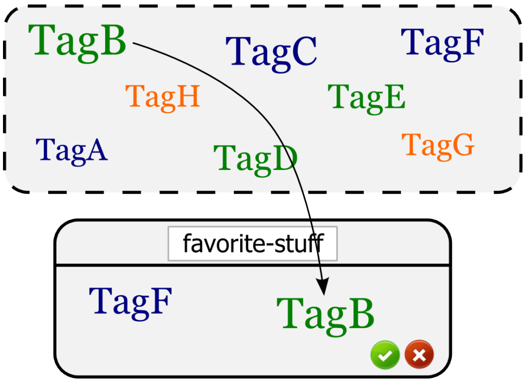


Figure 14: Create a tag bag

or to manually switch between contexts. Alternatively, the system can automatically switch between contexts.

The rationale behind the idea of associating tags to contexts is that some tags are often used or applied in certain contexts only. E.g. some tags might only be applied when doing daily business and working in the office, whereas others might be applied when travling (e.g. tags like ”weather_information” or ”traffic_information”). Of course tags needed depends on the context, too. So, the tags that have been applied when having been traveling might be irrelevant when being in office.

Thus always displaying all tags part of a tag space is often not reasonable. Hence, we allow for context-senstitive tag widgets, especially tag clouds that only display tags relevant in the current context.

3.8 Other concepts

We are also allowing for tag sharing among subcommunities. Most current tagging systems allow to either create public or private tags but do not allow for a granularity in between. Our prototype allows to share tags with a dedicated set of other users.

The tagging paradigms presented can be combined with one or more of the other paradigms presented. For example, an alien tag can be created which is only valid for a certain period of time.

4 Conclusion and Future Work

In this paper we have presented tagging paradigms which we are using to refine our user- and context modeling approaches presented in our previous work [Nauerz *et al.*, 2008] in order to perform content adaptation and recommendation. The concepts described have already been prototypically implemented and can be presented at the workshop. We have not yet performed in-depth evaluation on these early ideas described in this short paper but are looking forward to discuss them and receive initial feedback. Of course, especially the usefulness of each single concept has still to be evaluated.

For the future we plan to merge our Web 2.0 collaborative tagging approaches with Semantic Web ideas heading towards the Web 3.0.

IBM and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries or both. Other company, product and service names may be trademarks or service marks of others.

References

[Echarte *et al.*, 2009] Francisco Echarte, Jos Javier Astrain, Alberto Crdoba, and Jess E. Villadangos. Improving folksonomies quality by syntactic tag variations grouping. In Sung Y. Shin and Sascha Ossowski, editors, *SAC*, pages 1226–1230. ACM, 2009.

[Garg and Weber, 2008a] Nikhil Garg and Ingmar Weber. Personalized, interactive tag recommendation for flickr. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 67–74, New York, NY, USA, 2008. ACM.

[Garg and Weber, 2008b] Nikhil Garg and Ingmar Weber. Personalized tag suggestion for flickr. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 1063–1064, New York, NY, USA, 2008. ACM.

[Gwizdka and Bakelaar, 2009] Jacek Gwizdka and Philip Bakelaar. Tag trails: navigation with context and history. In *CHI EA '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, pages 4579–4584, New York, NY, USA, 2009. ACM.

[Lee and Han, 2007] Sung Eob Lee and Steve SanKi Han. Qtag: introducing the qualitative tagging system. In *HT '07: Proceedings of the eighteenth conference on Hypertext and hypermedia*, pages 35–36, New York, NY, USA, 2007. ACM.

[Liu *et al.*, 2009] Dong Liu, Xian-Sheng Hua, Linjun Yang, Meng Wang, and Hong-Jiang Zhang. Tag ranking. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 351–360, New York, NY, USA, 2009. ACM.

[Nauerz *et al.*, 2008] Andreas Nauerz, Stefan Pietschmann, and Rene Pietzsch. Social recommendation and adaptation in web portals. In *Proceedings of the International Workshop on Adaptation for the Social Web*, Hannover, Germany, 2008.

[Sen *et al.*, 2009a] Shilad Sen, Jesse Vig, and John Riedl. Learning to recognize valuable tags. In Cristina Conati, Mathias Bauer, Nuria Oliver, and Daniel S. Weld, editors, *IUI*, pages 87–96. ACM, 2009.

[Sen *et al.*, 2009b] Shilad Sen, Jesse Vig, and John Riedl. Learning to recognize valuable tags. In *IUI '09: Proceedings of the 13th international conference on Intelligent user interfaces*, pages 87–96, New York, NY, USA, 2009. ACM.

[Sigurbjörnsson and van Zwol, 2008] Börkur Sigurbjörnsson and Roelof van Zwol. Flickr tag recommendation based on collective knowledge. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 327–336, New York, NY, USA, 2008. ACM.

[Song *et al.*, 2008] Yang Song, Ziming Zhuang, Huajing Li, Qiankun Zhao, Jia Li, Wang-Chien Lee, and C. Lee Giles. Real-time automatic tag recommendation. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 515–522, New York, NY, USA, 2008. ACM.

[Suchanek *et al.*, 2008] Fabian M. Suchanek, Milan Vojnovic, and Dinan Gunawardena. Social tags: meaning and suggestions. In *CIKM '08: Proceeding of the 17th ACM conference* *on Information and knowledge management*, pages 223–232, New York, NY, USA, 2008. ACM.

[Symeonidis *et al.*, 2008] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 43–50, New York, NY, USA, 2008. ACM.

[Vig *et al.*, 2009] Jesse Vig, Shilad Sen, and John Riedl. Tagsplanations: explaining recommendations using tags. In *IUI '09: Proceedings of the 13th international conference on Intelligent user interfaces*, pages 47–56, New York, NY, USA, 2009. ACM.

[Wu *et al.*, 2006] Harris Wu, Mohammad Zubair, and Kurt Maly. Harvesting social knowledge from folksonomies. In *Proc. of the 17th Conf. on Hypertext and hypermedia*, pages 111–114, New York, NY, USA, 2006. ACM Press.

[Zhang *et al.*, 2009] Shaoke Zhang, Umer Farooq, and John M. Carroll. Enhancing information scent: identifying and recommending quality tags. In *GROUP '09: Proceedings of the ACM 2009 international conference on Supporting group work*, pages 1–10, New York, NY, USA, 2009. ACM.

Link Clouds and User-/Community-Driven Dynamic Interlinking of Resources

Andreas Nauerz and Martin Welsch
IBM Research and Development
71032 Böblingen, Germany
{andreas.nauerz|martin.welsch}
@de.ibm.com

Fedor Bakalov and Birgitta König-Ries
University Jena
07743 Jena, Germany
{fedor.bakalov|koenig}
@informatik.uni-jena.de

Abstract

During the last years we have observed a shift in the way how content is added to web-based systems. Earlier, dedicated authors were responsible for adding content, today entire communities contribute. As a consequence these systems grow quickly and uncoordinated. New ways had to be found to organize and structure content.

Tagging has become one of the most popular techniques to allow users (and entire user communities) to perform this structuring autonomously. But, not only because current tagging systems have their flipsides (e.g. synonyms and polysems lead to littered tag spaces making it difficult for users to find relevant content), we argue that tagging is sometimes an abstraction layer not necessarily needed. In many scenarios users just want to interlink content fragments (resources) with each other. In this paper we present an approach allowing users, i.e. the community, to collaboratively define relations between arbitrary content fragments. They can interlink any source with any target. We allow for personal interlinking of resources as well as collaborative interlinking. In the latter case we visualize, for each single resource, available interlinks in what we call link clouds, a concept comparable to tag clouds. We finally leverage the knowledge about the interlinks between resources' for building personal (or community) navigation structures and for performing content recommendations.

The concepts presented are being prototypically implemented within IBM's WebSphere Portal and can be presented in a live demo at the workshop.

1 Introduction

Today, web-based systems are often comprised of a huge amount of content. They are no longer exclusively maintained by IT departments, instead, Web 2.0 techniques are used increasingly, allowing user generated content to be added. These systems grow quickly and in a more uncoordinated way as different users possess different knowledge and expertise and obey to different mental models.

The continuous growth makes access to really relevant information difficult. Users need to find task- and role-specific information quickly, but face information overload and often feel *lost in hyperspace*. Thus, users often miss out on resources that are potentially relevant to their tasks, simply because they never come across them. On the one hand, users obtain too much information that is not relevant to their current task, on the other hand, it becomes cumbersome to find the right information and they do not obtain all the information that would be relevant.

As users (and entire communities) have been enabled to contribute content, mechanisms have been introduced to allow categorizing, organizing and structuring this content, too. Particularly tagging and rating, which have become very popular collaboration techniques, provide new means doing this kind of categorization. It can add valuable meta information and even lightweight semantics to web resources. Tagging allows non-expert users to develop folksonomies that categorize content available in the system.

In our previous work [Nauerz *et al.*, 2008] we developed several tagging engines, which e.g. allowed arbitrary annotators, e.g. human users or analysis components (for automated tagging), to annotate any resources. The analysis of users tagging behavior allowed us to model their interests and as well as semantic relations between resources, and thus to perform reasonable recommendations and adaptations. In [Nauerz *et al.*, 2009] we also introduced new tagging paradigms like alien tagging, reputation-based tagging, quantitative tagging, anti tagging, tag expiry, contextual tagging, and described how these can be used to refine our models and to perform even more valuable adaptations or to issue more valuable recommendations.

But tagging engines also have their flipsides, though: synonyms and polysems lead to littered tag spaces making it difficult for users to find relevant content. Users suffer from retrieving content actually not being of interest or, vice versa, from not retrieving content that actually would be of interest when exploring the tag space. Even worse, tagging requires users to invest work and thus time: they need to come up with proper tags and assign them to the appropriate resources. If users just want to interlink resources with each other this is an unnecessary overhead, probably one reason why in most tagging systems not more than approximately 20% of all users tag content (cp [Al-Khalifa and Davis, 2007] and [Sen *et al.*, 2006]).

In this paper we present an new approach for solving the problems just mentioned. We argue that if we allow people to contribute content, we should also allow them to organize and structure this content leveraging their collective wisdom. But we want to enable them to do so without forcing them to come up with proper tags for resources. We want to relief them from this overhead if not really necessary.

We regard tagging as an interesting approach to categorize content and see dynamic interlinking as an interesting accompanying approach to relate content fragments to each other.

Thus we present an approach allowing users, i.e. the community, to collaboratively define relations between arbitrary content fragments. They can interlink any source with any target. We allow for personal interlinking of resources as well as collaborative interlinking. In the latter case we visualize, for each single resource, available interlinks in what we call link clouds, a concept comparable to tag clouds. We finally leverage the knowledge about resources' interlinking for building personal navigation structures and for performing content recommendation.

2 Related Work

As already indicated, a lot of newer approaches to allow users to categorize, organize and structure content autonomous have been made by introducing abstraction layers like tagging.

But only few work has been done to find solutions allowing users to directly interlink resources. Even lesser work has been invested to find solutions leveraging knowledge about the interlinks created to aggregate link clouds (cp. 3.3), to construct personal- or create new navigation menus (cp. 3.4), or link flows (cp. 3.6), or to do content recommendations (cp. 3.5).

So far, most approaches rely on means to automatically improve link structures. Adapting link structures (including link sorting, link annotation, and link hiding as well as generating links) based on user profiles etc. has been performed a lot, approaches are e.g. described in [Brusilovsky, 1996]. Even earlier work on computed and adaptive linking included the implicit linking mechanism described by [DeRose, 1989], as well as the work described in [Bieber and Kimbrough, 1992] and [Stotts and Furuta, 1991].

Other early approaches to automatically interlink resources focus on computing links based on relationships or similarities between texts or passages of text, where a link is not defined as a pointer from one hypertext node to another, but rather as a query that leads to a different node. [Allan, 1996] describes how documents can be analyzed and automatically interlinked if similar. [Bodner and Chignell,] describes how text analysis can be performed on what they refer to as source nodes and target nodes. Depending on the similarity of both nodes links are automatically generated between those. A similar approach is described in [Wilkinson and Smeaton,] which is also based on the determination of the relationships between nodes to interlink them.

In [Nauerz and Welsch, 2007] we have described our approaches for automatically adapting link structures (and navigation topologies).

Some approaches to manually create links between resources (e.g. documents) are described in [Carr *et al.*, 1998]. But even with these approaches, where the community is given the power to decide which resource to interlink to which other resource, concepts like the ones mentioned above have not been pursued.

3 Concepts

Web-based systems are comprised of content fragments (also referred to as resources). These resources can be structuring elements like web pages, or with respect to Portal systems also pages and portlets. These resources provide users with content and services. On a more fine-granular basis resources can be any identifiable information unit, an image, a video, a document, a text passage, and so forth. Different resources provide different information, which can still be related. E.g., there might be pages part of an Enterprise Information Portal that provide means to book flights, hotels, cars or trains - different pages with similar use cases.

Prior to the Web 2.0 era these resources have been brought into relation by some central instances, usually administrators or content authors. However, those superimposed structures were not necessarily compliant to users' mental models and therefore resulted in significant effort to find the information needed. This became even worse, once user generated content was added, where the structure did not follow the design the administrator had in mind. fig. 1 shows the structure of a sample system: four branches exist below the root node. Along the first branch authors have put everything having something to do with ”flying”, e.g. pages that provide information about airports (location, arrival and departure times of flights, etc.), travel regulations (official regulations and internal company regulations), and finally a page to book a flight. Along the second branch authors have put everything have something to do with ”hotels”, e.g. pages that provide information about hotels at different locations as well as a page to finally book a hotel. Along the third branch authors have put similar pages having something to do with ”cars and trains”. Underneath the fourth branch users find pages to do their travel expense.

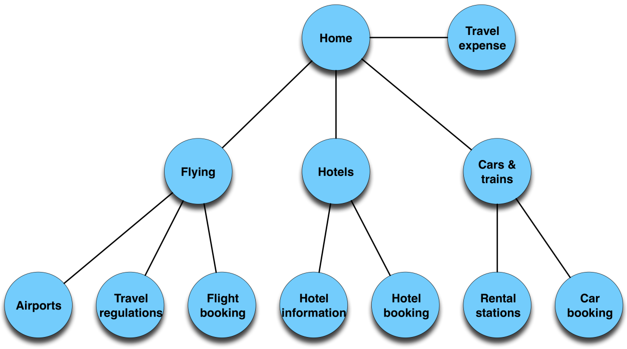


Figure 1: Structure of the sample system

Experienced users know about their favorite airports, the external and internal travel regulations and so forth - they just want to do their bookings. Given the structure above users would have to perform a lot of (unnecessary) clicks to traverse the *booking* pages. With the availability of a tagging engine users could have had tagged (which would have been work, too) these pages with the term ”booking”, but even then users would have to fire up the tag cloud, select the right tag, analyze the result list of resources that have been tagged with the selected tag and select the right one out of those being presented. Moreover, there could be more pages tagged with the term ”booking” but being irrelevant in the scenario described.

As said, the question is, why, if users contribute content, we do not allow them to organize and structure this content, too? Or, in other words, why we do not allow them to interlink resources independently from what administrators or content authors thought is the correct structure?

3.1 Personal Dynamic Interlinking

Private personal interlinks behave similar as private tags as described in [Nauerz *et al.*, 2009]. They can only be seen by the user who created them. fig. 2 exemplarily visualizes how private personal interlinks can be created within a typical Web Portal solution. First, the user navigates to the resource where he wants to interlink from, the source resource. A resource can be a page, a portlet or anything uniquely referencable. There he clicks a button which triggers the linking process. Next, he navigates to the resource he wants to interlink to, the target resource and clicks a button which finishs the linking process and establishes the interlink between both resources.

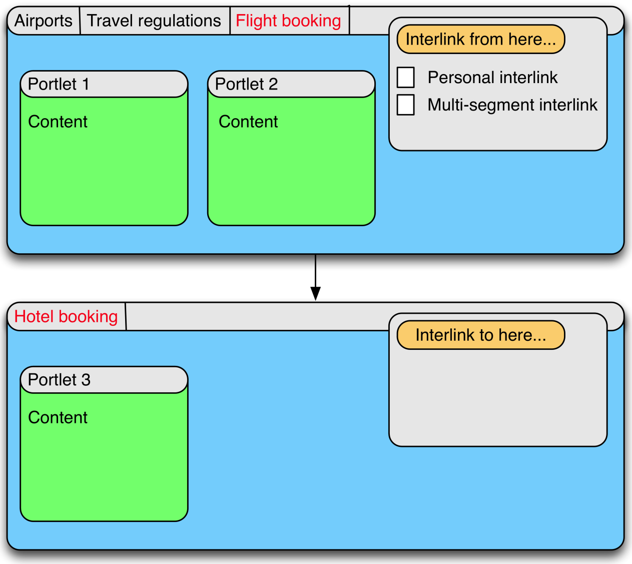


Figure 2: Creating interlinks

This way every user can interlink the resources he personally thinks should be related, totally independent from what an administrator or content author had in mind who always try to create structures satisfying majorities but not necessarily single users. He can manually create personal shortcuts and cross-references between related content. This way navigating through the system can be personalized and speed-up.

In the sample described earlier a user might be one of those experienced users that usually want to do his booking just by sequentially traversing the three *booking* pages and the *travel expense* page. Thus he would create three personal interlinks as depicted in fig. 3 (red connectors), one from the *flight booking* page to the *hotel booking* page, one from the *hotel booking* page to the *car booking* page, and one from the *car booking* page to the *travel expense* page. Next time he is doing his bookings he can follow this path by doing three clicks only, just following his personal interlinks.

3.2 Collaborative Dynamic Interlinking

The real power and benefits of dynamic interlinking becomes evident when allowing collaborative dynamic interlinking. A collaborative dynamic interlink created by one user can be seen by all other users, too. Creating collaborative interlinks is done similar as creating private interlinks, except that a checkbox indicating that the next interlink to be created should be a private interlink has not to be selected (cp. fig. 2).

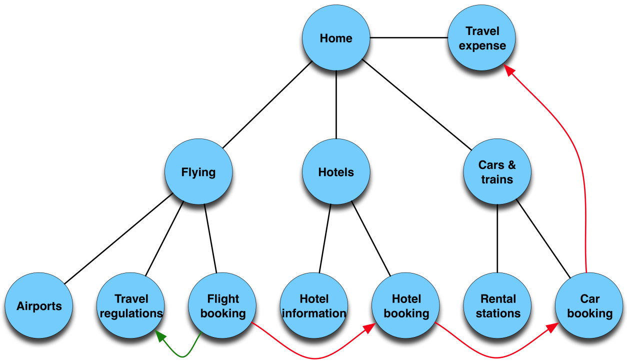


Figure 3: Personal interlinks

The private interlinks created before could have been created as collaborative interlinks, too. Thus, one or more experienced users could have set interlinks between the *booking* pages. This can help people that want to do their booking for the very first time. These users do not need to search for one *booking* page after the other anymore, instead they can follow the interlinks available.

Private and collaborative interlinks can be mixed, of course. E.g., in addition to the collaborative interlinks (red connectors) interlinking the *booking* pages, less experienced users might want to interlink from these pages to the corresponding pages providing information for travel regulations (green connectors) (cp. fig. 3 again).

3.3 Visualizing Dynamic Interlinks

An important aspect is that of course multiple interlinks can be created from any resource to any other; similar each single resource can be the endpoint of several interlinks pointing to it. Depending on users needs interlinks could also be added between other *booking* pages (cp. fig. 4 where solid as well as dotted red lines represent collaborative interlinks), e.g. between the *flight booking* page and the *car booking page*, the *flight booking* page and the *travel expense* page and so forth. This could be done by users that e.g. do never need all three booking pages, e.g. because they never book a car and want to skip the corresponding page.

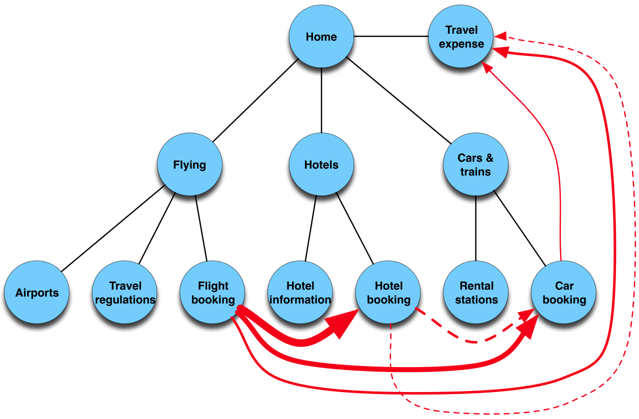


Figure 4: Collaborative interlinks

This is similar to what we observe in collaborative tagging environments, where single resources can be tagged with multiple tags. The most important concept to visualize tags (and their importance) for single resources or a set of resources are tag clouds. Tag clouds display which tags are available and how often these have been applied (with respect to what one is looking at); more often applied tags are regarded as more important tags which are usually presented in a larger font size.

We propose a similar mechanism, which we refer to as **link clouds**, for visualizing dynamic interlinks. When navigating from one resource to another, the user-/community created personal interlinks can be used in addition to the links that were originally created by an administrator or content author.

So, e.g. if collaborative interlinks have been created as described above, when being on the *flight booking* page there could be a link to the superior *flights* page as modeled by the content author. But additionally a link cloud could display interlinks to the *hotel* page, the *cars* page and the *travel expense* page. If most users navigate from the *hotel* page to the *cars* page more people would interlink these two pages. So, if e.g. 10 users interlink the *flight booking* page to the *hotel* booking page, 5 from the *flight booking* page to the *cars* booking page and 1 from the *flight booking* page to the *travel expense* page the first linkage would be regarded the most important one, the second one the second most important one and so forth. fig. 4 visualizes this as the thicker solid red connectors represent interlinks set by more users.

Link clouds visualize this importance to the users. Different solutions can be thought of. In one embodiment (cp. fig. 5) link clouds could look like tag clouds presenting a description of the target resource they are linking to. Depending on the importance of the available collaborative interlinks (derived by how often a certain interlink has been set) some targets could be presented more prominent (larger font size) than others.

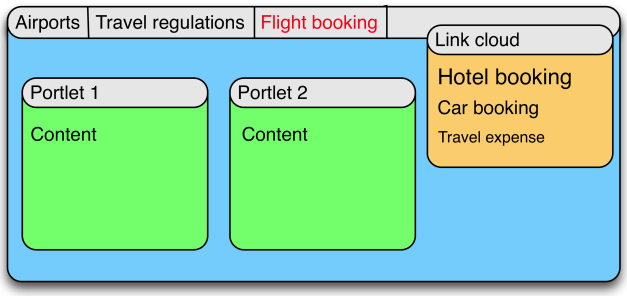


Figure 5: Link clouds

3.4 Personal and Community Navigation

Taking into consideration all personal and collaborative interlinks available in the system we can provide users with additional navigation menus, accompanying the one originally created by an administrator or content author, from which they can select. Therefore we provide users with a pull down menu, displayed at the top corner of every page, where he can select between these navigation menus:

- Original navigation
- Personal navigation
- Community navigation
- Aggregated navigation

The *original navigation* represents the navigation as created by an administrator or content author not containing any personal or community interlinks; The *personal navigation* adds personal interlinks to the *original navigation* so that these can be used from within the standard navigation menu. The *community navigation* adds collaborative interlinks to the *original navigation* and the *aggregated navigation* adds personal and collaborative interlinks to the *original navigation*.

It is also possible to display a navigation menu comprised of personal or community interlinks only, not containing the *original navigation* at all. This can be controlled via an additional check-box.

With respect to our previous sample, fig. 6 shows the *aggregated navigation*, which contains all administrator created links, as well as all collaborative interlinks and the user's personal interlinks. The same figure without the green connectors would represent the *community navigation*, without the red connectors the *personal navigation* and without the green and red connectors the *orgininal navigation*.

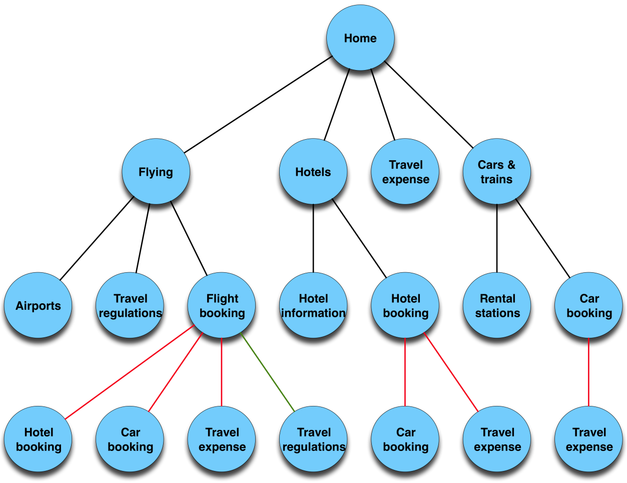


Figure 6: Personal-/community navigation

3.5 Content Recommendation

Leveraging the knowledge about incoming and outgoing dynamic interlinks for any resource allows us to perform related content recommendations. Three scenarios can be thought of (cp. fig. 7):

Forward linking (red lines in fig. 7) describes the most trivial case. With respect to our previous sample we might know that interlinks exist pointing from the *flight booking* page to the *hotel booking* page, *car booking* page and *travel expense* page. Thus we know that all these three target pages have something to do with the source page and can be recommended when being on the source page.

Backward linking (red lines in fig. 7) describes the second case. We might know that the *flight booking* page, the *hotel booking* page, and the *car booking* page link to the *travel expense* page. Thus we could recommend these three source pages when being on the target page.

Sideward linking (red lines in fig. 7 again) describes the third case. Again, we might know that interlinks exist pointing from the *flight booking* page to the *hotel booking* page, *car booking* page and *travel expense* page. Thus, there might not only be a relationship between the source and target pages, but also among the sources (or targets) themselves. Thus, a user being on the *hotel booking* page might also be interested in the *car booking* page as both are referenced from the same source page.

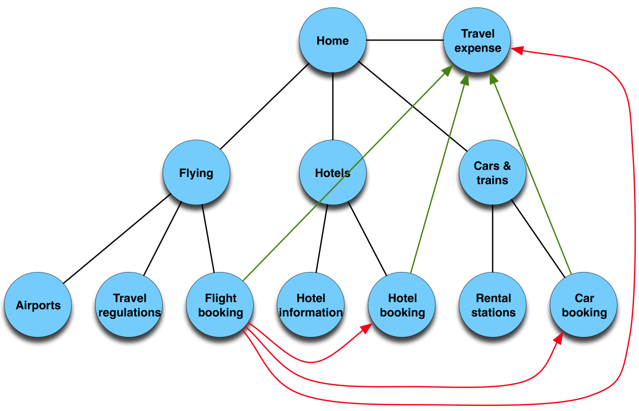


Figure 7: Forward, backward and sideward linking

3.6 Multi-Segment Interlinking

We also allow for doing more than just interlinking one resource to exactly one other. We refer to a continuous sequence of interlinks as **link flows**.

Such paths could be manually created by users (an additional check-box in the UI, cp. fig. 2, allows to do so), or, in a more sophisticated variant detected and recorded by the system. Latter could be based on following, from one resource to another, the ”top” interlink (the one set by most users), or on analyzing which available interlinks users follow, again from one resource to another, most often.

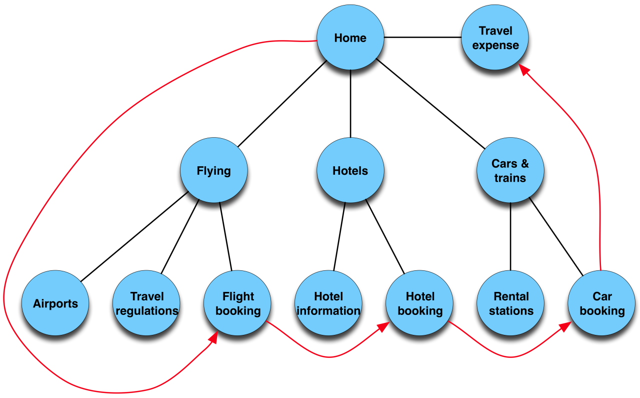


Figure 8: Multi-segment linking

With respect to our sample such a path could be comprised of the resources *home* page, *flight booking* page, *hotel booking* page, *car booking* page, and *travel expense* page (cp. fig. 8).

4 Conclusion and Future Work

In this paper we have presented a new approach to interlink arbitrary fragments of web elements by defining relations between them. Users can do this both only personally, but also in a collaborative way. Such relations can then be visualized and used for navigation purposes. For individual use such personal interlinks work like personal tags. Users can create their own optimized linking/relationship network independent from schemes defines by administrators or content authors. In the collaborative case the created link patterns are visible to all and provide added value by reflecting common or strongly used relations. We also proposed to eploit the explicitly generated knowledge about interlinking for content recommendation. Furthermore interlinking can be extended beyond single steps thus creating link flows or paths that could e.g. encompass an entire task to be performed. Users can always freely select between the original or the new enhanced link cloud navigation. These concepts have been prototypically implemented in IBM's WebSphere Portal.

In the future, we intend to work on visualization and UI related extensions. We are looking forward to discuss these ideas and get feedback. Each aspect still needs to be evaulated in terms of both usabilitiy and usefulness.

IBM and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries or both. Other company, product and service names may be trademarks or service marks of others.

References

[Al-Khalifa and Davis, 2007] Hend S. Al-Khalifa and Hugh C. Davis. Towards better understanding of folksonomic patterns. In *Proc. of the 18th Conf. on Hypertext and hypermedia*, pages 163–166, New York, NY, USA, 2007. ACM Press.

[Allan, 1996] James Allan. Automatic hypertext link typing. In *HYPERTEXT '96: Proceedings of the the seventh ACM conference on Hypertext*, pages 42–52, New York, NY, USA, 1996. ACM.

[Bieber and Kimbrough, 1992] Michael P. Bieber and Steven O. Kimbrough. On generalizing the concept of hypertext. *MIS Q.*, 16(1):77–93, 1992.

[Bodner and Chignell,] Richard Bodner and Mark Chignell. Dynamic hypertext: querying and linking. *ACM Comput. Surv.*, page 15.

[Brusilovsky, 1996] Peter Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.

[Carr *et al.*, 1998] L. A. Carr, D. De Roure, W. Hall, and G. Hill. Implementing an open link service for the world wide web. *World Wide Web*, 1(2):61–71, 1998.

[DeRose, 1989] S. J. DeRose. Expanding the notion of links. In *HYPERTEXT '89: Proceedings of the second annual ACM conference on Hypertext*, pages 249–257, New York, NY, USA, 1989. ACM.

[Nauerz and Welsch, 2007] Andreas Nauerz and Martin Welsch. (Context)Adaptive Navigation in Web Portals. In *Proc. of the Intl. IADIS WWW/Internet Conference*, Vila Real, Portugal, October 2007.

[Nauerz *et al.*, 2008] Andreas Nauerz, Stefan Pietschmann, and Rene Pietzsch. Social recommendation and adaptation in web portals. In *Proceedings of the International Workshop on Adaptation for the Social Web*, Hannover, Germany, 2008.

[Nauerz *et al.*, 2009] Andreas Nauerz, Fedor Bakalov, Martin Welsch, and Birgitta Knig-Ries. New tagging paradigms for content recommendation in web 2.0 portals. In *Proceedings of the International Workshop on Adaptation and Personalization for Web 2.0 (in conjunction with the 1st and 17th International Conference on User Modeling, Adaptation and Personalization)*, Trento, Italy, 2009.

[Sen *et al.*, 2006] Shilad Sen, Shyong K. Lam, Al Mamunur Rashid, Dan Cosley, Dan Frankowski, Jeremy Osterhouse, F. Maxwell Harper, and John Riedl. tagging, communities, vocabulary, evolution. In *Proc. of the 20th Conf. on Computer supported cooperative work*, pages 181–190, New York, NY, USA, 2006. ACM Press.

[Stotts and Furuta, 1991] P. David Stotts and Richard Furuta. Dynamic adaptation of hypertext structure. In *HYPERTEXT '91: Proceedings of the third annual ACM conference on Hypertext*, pages 219–231, New York, NY, USA, 1991. ACM.

[Wilkinson and Smeaton,] Ross Wilkinson and Alan F. Smeaton. Automatic link generation. *ACM Comput. Surv.*, page 27.

Adding Flexible Input Device Support to a Web Browser with MundoMonkey

Daniel Schreiber
TU Darmstadt
Darmstadt, Germany
schreibertk.informatik.tu-darmstadt.de

Abstract

Computer applications are increasingly used in non-desktop settings, e.g. at a public kiosk systems or on mobile phones. Thanks to the widespread availability of web browsers for different platforms, web interfaces are often employed in these settings. However, current browsers lack sufficient support for flexibly adapting to non-desktop settings, e.g. ad-hoc changes of input devices. A use case for this is, e.g., an interactive shopping window that presents a web interface for buying products. If the browser in the shopping window supported ad-hoc changes of input devices, the customer could dynamically attach carried input devices, e.g., a mobile phone, to the browser and interact with it. In this paper, we present a solution to the problem of dynamically connecting input devices in a non-desktop setting to a browser, based on the MundoMonkey Firefox extension for interactive spaces. Using our approach, the unmodified web user interface can be used with arbitrary input devices in ways that cannot be realized by synthesizing mouse and keyboard events. In our approach, the customization to the device at hand is performed transparently to the application developer by the end-user.

1 Introduction

Web interfaces are commonly used to make applications accessible for a wide range of usage situations. Web interfaces are extremely portable, e.g., they can be used on mobile phones, on public kiosk systems and on the office desktop. Still, there are many more situations, in which web interfaces cannot be used efficiently, because no browser exists that supports the situations characteristics with respect to in- and output devices.

If one such situation is important enough, e.g., because it is directly related to business value, a customized solution can be implemented for a controlled environment like a warehouse or a hospital following the design approach of [Klug and Mühlhäuser, 2007]. In more open environments, where a heterogeneous set of users interacts with different input devices in a way depending on their preferences and the situation, such an approach is not feasible. For example, in a shopping mall users may want to interact with an interactive shopping window using their personal devices for input, as sketched in Figure 1. The user's mobile phone can act as an input device, e.g., by using voice recognition via the built-in microphone or gesture recognition via the phone's camera or remotely controlling a mouse cursor via the phone's touchscreen. Thereby, the choice which modality is used for interaction depends on the situation, e.g., on the noise level and the user, e.g., her motor skill level.

In this paper, we argue that, ideally

- A the browser running the web interface, e.g., the one situated in an interactive shopping window, is able to automatically connect to and receives input from devices carried by the user, and
- B the user (or at least the shop assistant) can easily install support for novel interaction techniques and devices.

Reaching both subgoals should not require assistance from the developer of the web application nor assistance from the developer of the browser. We assume that in the envisioned open environments an existing interface will be used much more frequently in a novel situation or with a novel device than a new application for existing devices and situations will be developed. This is in contrast to desktop computing where an application would always be accessed with the same device setup. However, for web applications this is already true to some degree, as they may be accessed with different and novel browsers. Therefore, we put an emphasis on web user interfaces in our research, as these also bring many advantages, like easy deployment, and already address a wide range of usage situations as stated above. In this paper, we present a novel approach for connecting input devices to a web browser in an open environment, building on the MundoCore middleware [Aitenbichler *et al.*, 2007]. We integrated access to the middleware into an end-user scripting framework for web interfaces with the MundoMonkey extension for Firefox. Thereby,

- i) the actual interaction device used to carry out an interaction technique is determined automatically at runtime and
- ii) the set of supported interaction techniques can be determined by an end-user with minimal effort.

2 MundoCore Middleware for Interactive Spaces

To reach subgoal A, from the introduction, the different devices in an open environment, e.g., in front of a window in a shopping mall, need to be able to communicate with each other without configuration. A suitable approach to solve this problem, is to require that all devices use a common communication middleware like [Vanderhulst *et al.*, 2007] or [Johanson and Fox, 2002]. To this end, we

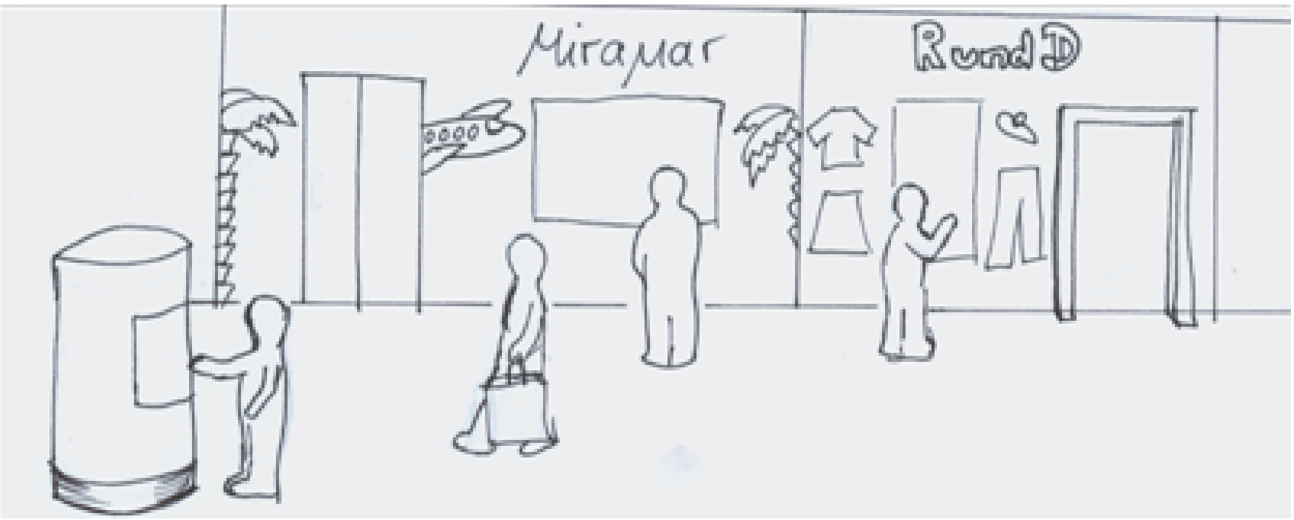


Figure 1: Users in a shopping mall interact with interactive shopping windows using different input devices.

employ the MundoCore middleware [Aitenbichler *et al.*, 2007]. MundoCore is a Pub/Sub middleware with language bindings for Java, Objective-C, C and C++. To connect a device to the MundoCore middleware a proxy service is needed that receives input events from the device hardware and publishes it as MundoCore events. Proxy services for many input devices are readily available, e.g., a voice recognition device [Aitenbichler *et al.*, 2004] or the Wii Remote. The ensemble of a hardware device and the corresponding MundoCore service publishing events will be called an *interaction resource* in this paper.

The ensemble of all interaction resources together with additional context sensors providing information build up the *interactive space* surrounding the browser. Thereby, we consider the interactive space as a generalization of desktop and mobile phone environments, for which current browsers are designed.

Several techniques can be used on the side of the browser to decide which of the available interaction resources should currently be used for interaction. One possibility is to decide based on location information, e.g., to accept input from devices belonging to a nearby user, or a user looking at the browser's screen, as proposed in [Braun *et al.*, 2004]. Another option, giving more control to the user, is a meta-user interface allowing the user to associate with the browser and configure the interaction means [Vanderhulst *et al.*, 2009; Schreiber and Hartmann, 2008]. Thus, the first subgoal A is reached by combining a communication middleware with context awareness and/or a suitable meta-user interface.

However, designing a browser for interactive spaces requires much more flexibility in supporting different input techniques, as the devices in the interactive space may change in unforeseen ways, e.g. requiring support for a so far unknown voice input device.

3 MundoMonkey

To reach the second subgoal B from the introduction, we designed the MundoMonkey Firefox extension. MundoMonkey [Schreiber *et al.*, 2009][1]. It takes the flexibility and rich output adaptation capabilities of end-user scripting for web interfaces and augments them with support for handling input from the interactive space surrounding the browser. Handling of input and output modifications are performed by MundoMonkey *interaction strategies*. MundoMonkey *interaction strategies* are JavaScript files that are executed in the context of the web application in the browser. This allows them to easily modify the output by operating on the DOM of the web page in the browser.

MundoMonkey is built on top of Greasemonkey[2] and Greasemonkey scripts can serve as basis for MundoMonkey *interaction strategies*. As one important improvement compared to Greasemonkey, MundoMonkey allows *interaction strategies* to connect to the MundoCore [Aitenbichler *et al.*, 2007] middleware for interactive spaces and thereby react to events from interaction resources and sensors in the interactive space. For example, *interaction strategies* can attach an event listener to a speech recognition service to react on user utterances.

The end-user can install a new *interaction strategy* like any Greasemonkey script, by pointing the browser to a URL. The set of available *interaction strategies* can thus be easily installed by the end-user, e.g., the shop assistant in the mall. The widespread use of Firefox extensions and Greasemonkey scripts shows in our opinion that installing *interaction strategies* is possible for the end-user without any assistance from developers.

At runtime, MundoMonkey automatically selects the appropriate *interaction strategies* for the devices at hand. MundoMonkey does so, by matching the device type to the required input for all *interaction strategies* and then selecting the ones that are best suited to handle the input from the devices. This setup can then be overridden by the user with the means of a meta-user interface.

Our approach requires that every *interaction strategy* is implemented in a generic way, i.e. able to work with any web user interface. Otherwise, installation of web user interface specific strategies would be needed, which would impose severe scalability problems considering the huge number of existing web user interfaces. Still, it is possible to support a wide range of input devices and interaction techniques by using MundoMonkey strategies. In

[1] https://leda.tk.informatik.tu-darmstadt.de/cgi-bin/twiki/view/Mundo/MCFirefox

[2] http://www.greasepot.net

the next section, we present one example of such an interaction strategy. Especially interaction techniques that cannot be realized by synthesizing mouse and keyboard events, e.g., interaction techniques that require the combination of output modification and input handling, benefit from MundoMonkeys output modification capabilities.

4 Voice Interaction Strategy

As a use case for MundoCore and MundoMonkey, we present an interaction strategy for voice. Thereby, the voice input device connected to the Firefox browser is the Talking Assistant [Aitenbichler *et al.*, 2004]. The Talking Assistant is a small device that is always carried by the user. The idea is, that the user in a mall can use the voice recognition capabilities of the Talking Assistant with a browser in an interactive shopping window. If somebody wearing a Talking Assistant approaches the shopping window, the Talking Assistant is connected to the browser and can control the web page loaded in the browser via voice commands.

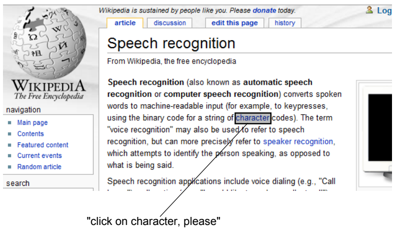
The problem solved by the voice interaction strategy is to provide a recognition grammar for the web page at hand to the Talking Assistant. This has to be done without any specific knowledge about the web application, as otherwise a new strategy for every interface would be required. Thanks, to the access to the DOM of the web interface in the browser, the strategy can be implemented in a web interface agnostic way.

The recognition grammar provides phrases for controlling every interactive element of the web page, extracted from the DOM tree. Thereby, the content of the element is padded with fixed phrases for making the interaction more natural. To access a link on the page, the user can e.g. say ”click on <link text>, please”. For interactive form fields the fieldname is not so easy to extract. We used existing algorithms for determining the labels of interactive elements on the web page, [Leshed *et al.*, 2008; Hartmann *et al.*, 2008]. Once the page is loaded and processed, the resulting grammar is sent to the Talking Assistant via the MundoMonkey Extension. Figure 3 shows an example for a rule generated for a link with the label ”character”, more complicated rules may also involve variables, i.e. inline dictation, whose values are passed back to the strategy as parameters. The grammar is specified in the MS SAPI5.3 format[3].

Once the grammar was sent to the Talking Assistant, the reactive part of the strategy handles all recognized utterance. To do so, it stores a specific callback function for every grammar rule. The callback functions are generated while processing the page and creating the grammar. For example, matching of the rule in figure 3 results in following the ”character” link in the web page. Our voice interaction strategy supports different actions for the different HTML form element types and HTML links.

5 User Study

To evaluate whether our voice interaction strategy provided usability which is comparable to other state of the art voice user interface techniques that do not benefit from the flexibility of MundoCore and MundoMonkey, we compared it against the built-in voice control of the Internet Explorer in Windows Vista.



Wikipedia is sustained by people like you. Please donate today.

Speech recognition

From Wikipedia, the free encyclopedia

Speech recognition (also known as **automatic speech recognition** or **computer speech recognition**) converts spoken words to machine-readable input (for example, to keypresses, using the binary code for a string of character codes). The term "voice recognition" may also be used to refer to speech recognition, but can more precisely refer to speaker recognition, which attempts to identify the person speaking, as opposed to what is being said.

Speech recognition applications include voice dialing (e.g., "Call

"click on character, please"

Figure 2: Example page used in the user study. The character link could be activated by saying ”Click on character, please”.

```
<rule id="command4">
  <O><O>please</O>
    <L>
      <P>select</P>
      <P>click<O>on</O></P>
    </L>
  </O>
  <P>character</P><O>please</O>
</rule>
```

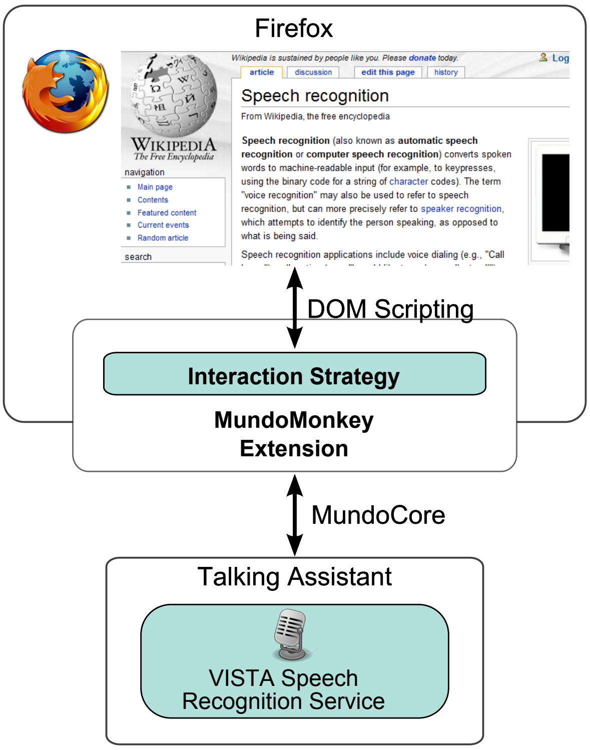
Figure 3: One rule of the grammar generated for the example page.

The study was conducted using a within subject design. Participants were members of our department and students ($n = 10$). Every participant completed a task with the Firefox browser, augmented with our voice input strategy extension and the Vista voice control for Internet Explorer. Thereby the Talking Assistant also used the Vista Speech Recognizer, so we reduced the difference to just the mapping of speech recognition results to actions in the web page. This procedure allowed us to test our interaction strategy against the built-in Internet Explorer strategy. See Figures 5 and 4 for an overview on the setup in both conditions. Although we tested only one web user interface in the user study, the strategy works with other websites, e.g., the one of E-Bay or of online travel-agencies. Thereby the example shows, that a strategy can be efficiently implemented without tailoring it to a specific website.

The task performed by the participants was to gather information from Wikipedia articles, which could be easily replaced by the contents of a shopping catalog in the mall scenario. Participants had to scroll down twice (they were told to ”Find the information on the bottom of the page”). Then they were instructed to select a certain link (”follow the third link in the list”) and then select a link of their choice (”follow any link on this page that interests you”). After the task, participants filled out the SUS usability questionnaire [Brooke, 1996]. The order of conditions was counterbalanced to control for learning effects. We found the ratings of our strategy ($M = 62.5, SD = 17.16$) were significantly better than the rating for the Vista Voice Interaction ($M = 51.5, SD = 18.33$) using a dependent samples t-test ($t(9) = 2.45, p < .05$), see figure 6. The effect size was medium to large with $Cohens'd = 6.4$.

The goal of the study was not to prove superior usability of the voice interaction strategy but more whether voice interaction could be implemented within MundoMonkey with comparable usability to commercial systems. Therefore we did not use a larger sample size or obtain detailed quantitative data. The reason why the voice strategy of MundoMonkey achieved a higher SUS score is probably the higher recognition rate of the Vista Speech Recognizer

[3] http://msdn.microsoft.com/en-us/library/ms723632(VS.85).aspx



Wikipedia is sustained by people like you. Please donate today.

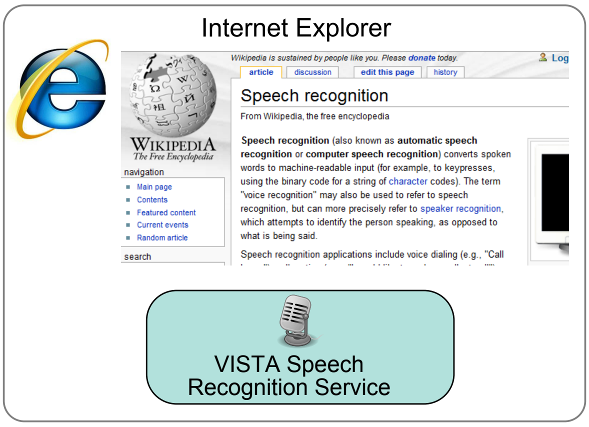
Speech recognition

From Wikipedia, the free encyclopedia

Speech recognition (also known as **automatic speech recognition** or **computer speech recognition**) converts spoken words to machine-readable input (for example, to keypresses, using the binary code for a string of character codes). The term "voice recognition" may also be used to refer to speech recognition, but can more precisely refer to speaker recognition, which attempts to identify the person speaking, as opposed to what is being said.

Speech recognition applications include voice dialing (e.g., "Call

Figure 4: Setup in the voice interaction strategy condition used in the study



Wikipedia is sustained by people like you. Please donate today.

Speech recognition

From Wikipedia, the free encyclopedia

Speech recognition (also known as **automatic speech recognition** or **computer speech recognition**) converts spoken words to machine-readable input (for example, to keypresses, using the binary code for a string of character codes). The term "voice recognition" may also be used to refer to speech recognition, but can more precisely refer to speaker recognition, which attempts to identify the person speaking, as opposed to what is being said.

Speech recognition applications include voice dialing (e.g., "Call

Figure 5: Setup for the builtin Internet Explorer strategy condition used in the study

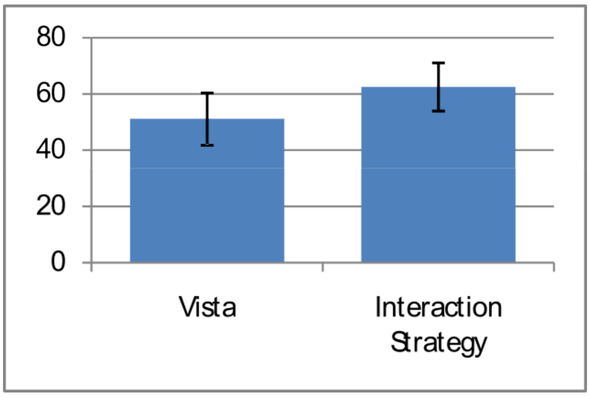


Figure 6: Our voice interaction strategy achieved a signaficantly higher SUS usability score compared to the builtin Internet Explorer strategy.

when using a small recognition grammar for the website at hand (as in the voice interaction strategy) compared to the dictation grammar used in the Vista condition.

Implementing voice interaction as *interaction strategy* within MundoMokey has several advantages over embedding it in the operating system or the browser, as e.g. done in Windows Vista. The recognition is done on the Talking Assistant, which exclusively is used by a single user. This allows the voice recognition engine of the Talking Assistant to be highly customized for a single speaker, greatly improving recognition performance. Further, the flexibility of MundoCore allows different Talking Assistants to dynamically associate with the browser and interact with the web application, as required in a mall which is populated by many users.

6 Related Work

User interfaces that adapt to the context of use, i.e. the interaction devices, the user experience ant the user's other tasks [Calvary *et al.*, 2005] without having to change their implementation through programming are a pertinent problem in research. In this section approaches to solving the problem from various fields will be analyzed according to their strength and weaknesses.

W3C Ubiquitous Web Applications Activity The problem of widening access to web applications to other devices and modalities is targeted by the W3C Ubiquitous Web Applications Activity which *focuses on mechanisms to reduce the cost for developing and delivering applications to a wide range of devices, including the means to adapt to user preferences and environmental conditions*. This goal is pursued by proposing extensions to the HTML standard, e.g. X+V. With such an approach, legacy applications, e.g. an existing travel booking web application, will have to be rewritten to benefit from these extensions. Supporting emerging input techniques will require an additional rewrite of the application. An alternative would be to map the new input techniques to fit an existing application in the browser, as it is proposed in this paper.

Model Driven UI Development Model driven UI development as e.g., proposed in [Berti *et al.*, 2004] allows the application developer to specify the UI in a device- or modality independent user interface description language (UIDL, see [Souchon and Vanderdonckt, 2003] for an overview of existing UIDLs) at an abstract level. This description is then automatically adapted to the device at hand. This approach gives the programmer very powerful tools to specify user interfaces. As noticed in [Gilroy and Harrison, 2005], it forces end-users and device vendors to integrating new devices or interaction techniques into the model transformation engine before they can be used, which is rather complex. As an alternative, we are focussing on allowing the end-user to customize the interaction with a web application in an ad-hoc easy way, thereby not giving the programmer additional support for specifying the interface.

Adaptive Toolkits Adaptive toolkits like SUPPLE++ [Gajos *et al.*, 2007] layout GUIs with respect to the physical abilities of the user or the requirements of the situation (e.g. to support users wearing gloves). SUPPLE++ relies on a proven mathematical model, which is able to automate the layout task using recorded user traces and an interaction cost model describing the situation. Implementing interaction techniques, like voice interaction or techniques which require modification of the UI as a whole instead of at widget level are more difficult to implement with such an approach. Also, the focus is not on supporting dynamically changing input devices at runtime, which would require to connect the adaptive toolkit to a suitable communication middleware.

Assistive Computing The above mentioned approaches provide advanced tools to the developer, which require a rewrite of legacy applications to benefit the end-user. Approaches from the area of assistive computing like [Wang and Mankoff, 2003; Carter *et al.*, 2006] do not require a change of the application to support a specific usage context. Instead they mimic the expected environment to the application and map it to the actual environment, e.g. by controlling a mouse cursor with a one-switch device. However, they do not support highly dynamic interactive spaces as e.g. encountered in the mall scenario, where one application is used by different users with different devices.

Ubiquitous Computing Approaches from the area of ubiquitous computing that explicitly target dynamically changing devices like [Dragicevic and Fekete, 2004b; Serrano *et al.*, 2008; Ballagas *et al.*, 2003; Dragicevic and Fekete, 2004a] do not work together with HTML interfaces, thereby losing the advantages web applications already provide, like easy deployment. Although one could theoretically implement a web browser which makes use of these approaches, supporting some interaction techniques would still be difficult. For example, the mentioned tools do not support output modification sufficiently. For recognition based input, like speech recognition devices this is a drawback as these are best handled by providing disambiguation options to the user, e.g., in the form of a list to choose from [Mankoff *et al.*, 2000]. In our approach output adaptation and presenting disambiguation lists is easily possible. The results from [Hartmann and Schreiber, 2009] show, that displaying suggestions from uncertain contextual data sources allows to increase usability of applications.

End-user scripting Altering web pages in the browser is supported by end user programming tools like [Bolin *et al.*, 2005; Little and Miller, 2006; Bigham and Ladner, 2007]. This approach allows the end user to tailor interaction with web pages to her specific situation without requiring to rewrite the application. However, these programming environments do not support communication with external input devices at the client side. This drawback is remedied by MundoMonkey, which thus enables to use the very successful end-user scripting approach for web applications to target the adaptation to input devices as well.

7 Conclusions

Web interfaces are widely used to provide access to applications in many non-desktop usage situations. However, current browsers are too inflexible to support the different situations we encounter in open environments, like a shopping mall. To remedy this, we presented a solution relying on i) MundoCore for dynamically connecting input devices to the browser at runtime, and ii) MundoMonkey for letting the end-user add support for new types of input devices in the form of interaction strategies without requiring changes to existing applications.

As an example we presented an implementation of voice input as interaction strategy, which has significantly better usability than the built-in voice control for Internet Explorer. Additionally, it can easily connect to different input devices, e.g., in a mall environment. Compared to other approaches, MundoMonkey allows the implementation of *interaction strategies* that handle the input to applications and additionally are able to modify the output of the application. For example, this makes it possible to combine voice input with suggestions from contextual data, as implemented in [Hartmann and Schreiber, 2009].

Currently, the interaction strategies need to be installed manually, however, they could also be downloaded automatically from the device of the end-user, which would further simplify the process.

References

[Aitenbichler *et al.*, 2004] Erwin Aitenbichler, Jussi Kangasharju, and Max Mühlhäuser. Talking Assistant: A Smart Digital Identity for Ubiquitous Computing. In *Advances in Pervasive Computing*, pages 279–284. Austrian Computer Society (OCG), Austrian Computer Society (OCG), 2004.

[Aitenbichler *et al.*, 2007] Erwin Aitenbichler, Jussi Kangasharju, and Max Mühlhäuser. MundoCore: A Lightweight Infrastructure for Pervasive Computing. *Pervasive and Mobile Computing*, 2007.

[Ballagas *et al.*, 2003] Rafael Ballagas, Meredith Ringel, Maureen Stone, and Jan Borchers. istuff: a physical user interface toolkit for ubiquitous computing environments. In *Proceedings of CHI*, 2003.

[Berti *et al.*, 2004] Silvia Berti, Francesco Correani, Giulio Mori, Fabio Paternò, and Carmen Santoro. Teresa: a transformation-based environment for designing and developing multi-device interfaces. In *Proceedings of CHI*, 2004.

[Bigham and Ladner, 2007] Jeffrey P. Bigham and Richard E. Ladner. Accessmonkey: a collaborative scripting framework for web users and developers. In *Proceedings of the international cross-disciplinary conference on Web accessibility (W4A)*, 2007.

[Bolin *et al.*, 2005] Michael Bolin, Matthew Webber, Philip Rha, Tom Wilson, and Robert C. Miller. Automation and customization of rendered web pages. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 163–172, Seattle, WA, USA, 2005. ACM.

[Braun *et al.*, 2004] Elmar Braun, Gerhard Austaller, Jussi Kangasharju, and Max Mühlhäuser. Accessing web applications with multiple context-aware devices. In Maristella Matera and Sara Comai, editors, *Engineering Advanced Web Applications*, pages 353–366. Rinton Press, December 2004.

[Brooke, 1996] J. Brooke. Sus-a quick and dirty usability scale. *Usability Evaluation in Industry*, pages 189–194, 1996.

[Calvary *et al.*, 2005] Galle Calvary, Jolle Coutaz, Olfa Dassi, Lionel Balme, and Alexandre Demeure. *Towards a New Generation of Widgets for Supporting Software Plasticity: The Comet*, pages 306–324. 2005.

[Carter *et al.*, 2006] Scott Carter, Amy Hurst, Jennifer Mankoff, and Jack Li. Dynamically adapting guis to diverse input devices. In *Proceedings of Assets*, 2006.

[Dragicevic and Fekete, 2004a] Pierre Dragicevic and Jean-Daniel Fekete. The input configurator toolkit: towards high input adaptability in interactive applications. In *Proceedings of AVI*, 2004.

[Dragicevic and Fekete, 2004b] Pierre Dragicevic and Jean-Daniel Fekete. Support for input adaptability in the icon toolkit. In *Proceedings of ICMI*, 2004.

[Gajos *et al.*, 2007] Krzysztof Z. Gajos, Jacob O. Wobbrock, and Daniel S. Weld. Automatically generating user interfaces adapted to users' motor and vision capabilities. In *Proceedings of UIST*, 2007.

[Gilroy and Harrison, 2005] Stephen W. Gilroy and Michael D. Harrison. Using interaction style to match the ubiquitous user interface to the Device-to-Hand. In *Engineering Human Computer Interaction and Interactive Systems*, pages 325–345, 2005.

[Hartmann and Schreiber, 2009] Melanie Hartmann and Daniel Schreiber. Augur: Providing context-aware interaction support. In *Engineering Interactive Computing Systems (EICS'09)*, Carnegie Mellon University, Pittsburgh, USA, 2009.

[Hartmann *et al.*, 2008] Melanie Hartmann, Torsten Zesch, Max Mühlhäuser, and Iryna Gurevych. Using similarity measures for context-aware user interfaces. In *Proceedings of 2nd International Conference on Semantic Computing*, page (to appear). IEEE, IEEE, August 2008.

[Johanson and Fox, 2002] Brad Johanson and Armando Fox. The event heap: A coordination infrastructure for interactive workspaces. In *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications*, 2002.

[Klug and Mühlhäuser, 2007] Tobias Klug and Max Mühlhäuser. Modeling human interaction resources to support the design of wearable multimodal systems. In *ICMI '07: Proceedings of the ninth international conference on Multimodal interfaces*, pages 299–306, New York, NY, USA, 2007. ACM.

[Leshed *et al.*, 2008] Gilly Leshed, Eben M. Haber, Tara Matthews, and Tessa Lau. Coscripter: automating & sharing how-to knowledge in the enterprise. In *Proceeding of CHI*, pages 1719–1728, 2008.

[Little and Miller, 2006] Greg Little and Robert C. Miller. Translating keyword commands into executable code. In *Proceedings of UIST*, pages 135–144, Montreux, Switzerland, 2006. ACM.

[Mankoff *et al.*, 2000] Jennifer Mankoff, Scott E. Hudson, and Gregory D. Abowd. Providing integrated toolkit-level support for ambiguity in recognition-based interfaces. In *Proceedings of CHI*, 2000.

[Schreiber and Hartmann, 2008] Daniel Schreiber and Melanie Hartmann. Association: Unobtrusively creating digital contracts with smart products. In *Proceedings of Smart Products: Building Blocks of Ambient Intelligence (AmI-Blocks'08)*, 2008.

[Schreiber *et al.*, 2009] Daniel Schreiber, Melanie Hartmann, and Max Mühlhäuser. Mundomonkey: Customizing interaction with web applications in interactive spaces. In *Engineering Interactive Computing Systems (EICS'09)*, Carnegie Mellon University, Pittsburgh, USA, 2009.

[Serrano *et al.*, 2008] Marcos Serrano, Laurence Nigay, Jean-Yves L. Lawson, Andrew Ramsay, Roderick Murray-Smith, and Sebastian Denef. The openinterface framework: a tool for multimodal interaction. In *Proceedings of CHI*, 2008.

[Souchon and Vanderdonckt, 2003] Nathalie Souchon and Jean Vanderdonckt. A Review of XML-compliant User Interface Description Languages. In *Proceedings of International Workshop Design, Specification, and Verification of Interactive Systems*, 2003.

[Vanderhulst *et al.*, 2007] Geert Vanderhulst, Kris Luyten, and Karin Coninx. Middleware for ubiquitous service-oriented spaces on the web. In *AINAW '07: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, pages 1001–1006, Washington, DC, USA, 2007. IEEE Computer Society.

[Vanderhulst *et al.*, 2009] Geert Vanderhulst, Daniel Schreiber, Kris Luyten, Max Mühlhäuser, and Karin Coninx. Edit, inspect and connect your surroundings: A reference framework for meta-uis. In *Engineering Interactive Computing Systems (EICS'09)*, Carnegie Mellon University, Pittsburgh, USA, 2009.

[Wang and Mankoff, 2003] Jingtao Wang and Jennifer Mankoff. Theoretical and architectural support for input device adaptation. In *Proceedings of CUU*, 2003.

Author Index

WIR 2009

Workshop Information Retrieval 2009

Editors
Thomas Mandl, Universität Hildesheim
Ingo Frommholz, University of Glasgow

Workshop Information Retrieval
WIR 2009

Ingo Frommholz
University of Glasgow
United Kingdom

Thomas Mandl
Universität Hildesheim
Germany

The IR Workshop

The ubiquity of search systems has led to the application of information retrieval technology in many new contexts (e.g. mobile and international) and for new object types (products, patents, music). In order to develop appropriate products, basic knowledge on information retrieval needs to be revisited and innovative approaches need to be taken, for example by allowing for user interaction or by taking the user's situational context into account. The quality of information retrieval needs to be evaluated for each context. Large evaluation initiatives respond to these challenges and develop new benchmarks.

The workshop Information Retrieval 2009 of the Special Interest Group for Information Retrieval within the German Gesellschaft für Informatik (GI) provides a forum for scientific discussion and the exchange of ideas. The workshop takes place in the context of the LWA "Learning, Knowledge and Adaptivity" workshop week (LWA, Sep. 21-23, 2009) at the Darmstadt University of Technology in Germany. It continues a successful series of conferences and workshops of the special interest group on information retrieval (`http://www.fg-ir.de`). The workshop addresses researchers and practitioners from industry and universities. Especially doctorate and master students are encouraged to participate.

WIR 2009

The following types of submissions were requested this year:

- Full Papers (9 accepted submissions)
- Short Papers (5 accepted submissions): Position papers or work in progress
- Poster and Demonstrations (3 accepted submissions): Poster and presentation of systems or prototypes

The following areas were covered by the workshop:

- Development and optimization of retrieval systems
- Retrieval with structured and multimedia documents
- Evaluation and evaluation research
- Text mining and information extraction
- Multilingual systems
- Digital libraries
- User interfaces and user behavior
- Interactive IR
- Combination of structured and unstructured search
- Machine learning in information retrieval
- Information retrieval and knowledge management
- Information retrieval and the semantic web
- Search Engine Optimization
- Social Search

Program committee

The program committee had the following members (in alphabetical order):

- Prof. Dr. Maximilian Eibl, TU Chemnitz
- Dr. Ingo Frommholz, University of Glasgow, UK (Chair)
- Prof. Dr. Norbert Fuhr, Universität Duisburg-Essen
- René Hackl, FIZ Karlsruhe
- Prof. Dr. Matthias Hemmje, Fernuniversität Hagen
- Prof. Dr. Andreas Henrich, Universität Bamberg
- Frank Hopfgartner, University of Glasgow, UK
- Dr. Claus-Peter Klas, Fernuniversität Hagen
- Dr. Michael Kluck, Stiftung Wissenschaft und Politik, Berlin
- Prof. Dr. Gerhard Knorz, Fachhochschule Darmstadt
- Sascha Kriewel, Universität Duisburg-Essen
- Prof. Dr. Reginald Ferber, Hochschule Darmstadt
- Prof. Dr. Joachim Griesbaum, Universität Hildesheim
- Dr. Johannes Leveling, Dublin City University, Ireland
- Dr. Thomas Mandl, Universität Hildesheim (Chair)
- Dr. Wolfgang Müller, EML Research
- Prof. Dr. Marc Rittberger, DIPF, Frankfurt am Main
- Prof. Dr. Vivien Petras, Humboldt-Universität, Berlin
- Dr. Thomas Roelleke, Queen Mary University of London, UK
- Prof. Dr. Stefan Rüger, Open University, Milton Keynes, UK
- Dr. Ralf Schenkel, Universität des Saarlandes, Saarbrücken
- Prof. Dr. Hinrich Schütze, Universität Stuttgart
- Prof. Dr. Ingo Schmitt, TU Cottbus
- Prof. Dr. Benno Stein, Universität Weimar
- Dr. Ulrich Thiel, Fraunhofer IPSI, Darmstadt
- Judith Winter, J.W. Goethe-Universität Frankfurt

- Prof. Dr. Christian Wolff, Universität Regensburg
- Prof. Dr. Christa Womser-Hacker, Universität Hildesheim

We would like to thank the authors for their submissions, and we also thank the members of the program committee for providing helpful constructive reviews.

Darmstadt, September 2009,

Thomas Mandl and Ingo Frommholz

Table of Contents

On Table Extraction from Text Sources with Markups

Lorenz Weizsäcker Johannes Fürnkranz
Technische Universität Darmstadt
{lorenz,juffi}@ke.tu-darmstadt.de

Abstract

Table extraction is the task of locating tables in a document and extracting their content along with its arrangement within the tables. The notion of tables applied in this work excludes any sort of meta data, e.g. only the entries of the tables are to be extracted. We follow a simple unsupervised approach by selecting the tables according to a score that measures the in-column consistency as pairwise similarities of entries where separator columns are also taken into account. Since the average similarity is less reliable for smaller tables this score demands a levelling in favor of greater tables for which we make different propositions that are covered by experiments on a test set of HTML documents. In order to reduce the number of candidate tables we use assumptions on the entry borders in terms of markup tags. They only hold for a part of the test set but allow us to evaluate any potential table without referring to the HTML syntax. The experiments indicate that the discriminative power of the in-column similarities is limited but also considerable given the simplicity of the applied similarity functions.

1 Introduction

This work is about table extraction on HTML-documents or on other source types that use mark-up-tags and can easily be transformed into a sequence of ASCII symbols without impeding the extraction of tables. As output we target the entries along with their arrangement in row and columns letting aside any sort of meta data.

Following [6], [3] the task of *table extraction* strips downs to two subtasks. For solving the task of *table location* we have to find the smallest substrings of the input string that contain the tables. For *table recognition* we aim to get the tables content and structure out of those substring having the input string at hand. A substrings is here defined by both, the sequence of character it contains and its *slice*, the start and the end position, on the string it is taken from.

In principle, mark-up-languages allow addressing tables with tools for querying the structure induce by the markups. However, for neither of the two subtasks we can fully rely on the markups. For instance, in HTML-documents *table*-nodes often are used for arbitrary layout porpuses. Further, the location task also has a significant genuine error in the sense that for some potential tables it is a matter of tast wether we takes them as extractioin targets or not. The correct output is not defined but by the choosen example set. We do not intend to extract meta data such as titles, column headers, separating rows that are empty or contain titles of the subsequent table part. Here again, for many types of meta there are special node definitions but these are not always used. For instance, we have often to deal with column headers which take, according the markups, the place of first data row of the table.

For the approach presented here we do not consider the definitions of the mark-up-languages. Instead, we confine ourself to inspecting statistics on the columns of a table where column refers to column of table entries and column of entry separators as well. More precisely, we measure the consistency of a column by the average pairwise similarity of its elements where similarity refers to a plug-in string kernel. The basic observation is that the entries in a given column of a table tend to be more similar to each other than to other potential table entries in the document [14],[2]. With this report we want to frame out an extraction algorithm that uses the in-column similarities as single extraction criterion and see to what extend the extraction of tables can draw thereon.

Our algorithm intends to consider all potential tables within the input string likewise searching substrings with certain properties. Without further restriction of the output space, this is too expensive since we do not only have to find the substrings of the input covering the tables but also determine therein all substrings representing the entries. If we regard any position in the string after a given position as candidate start point of the next entry, there are far too many candidate tables. For this reason, we consider documents with mark-up tags such HTML documents. Obviously, the search space for these documents is smaller since the border of a table entry can be assumed to concur with the border of a tag.

Unfortunately, real world documents contain thousands of tags such that the search space still is to big for inspecting it element by element. This means that we either have to use non-exhaustive search, e.g. we evaluate the extraction criterion only for some elements of the space and try to find the most promising candidates nevertheless. Or, we further reduce the search space by means of stronger assumptions on the entry borders. We have chosen latter option by applying the assumptions given in section 3 which shrink the size of the space of candidate outputs to a moderate size. The drawback of this choice, as reported in the experiments section 6, is that the assumptions hold only for a minority of the documents in the data set we used. We decided to take this loss in relevance of the results because at first we wanted elaborate the extraction criterion itself.

And indeed, there is a principle catch with extraction criteria such as the criterion specified in section 4.2 and 4.3. We intend to evaluate a candidate table by the average in-column similarity of the table entries. However, the reliability of this average score strongly suffers when the candidate table is small and we therefor average over a small number of similarities only. We respond to this problem by deliberately decreasing the chances of smaller candidate tables, a procedure to which we refer to as *levelling*.

The outline of this paper is the following. A review on related work is given in section 2. Section 3 provides notation and specifies the assumptions for the reduction of the search space. The raw extraction criterion, called table score, is given in section 4 while different proposals for levelling schemes are discussed in the section 5. In section 6 we report on experimental results based on the set data set from [14] and section 7 contains conclusions and depicts further work.

2 Related Work

2.1 Binary Classification with Content Features

The reference most related to this work is [14]. The authors reduce the table location task to a binary classification problem. First, the leaf table nodes are marked as candidate target tables. Then, each candidate table is mapped to a feature vector that in turn is the input to a classifier which decides whether the vector represents a target table or not. The feature vector consists of the feature groups for words, layout and content respectively.

For *word features* a bag of words is collected from the text nodes of each candidate table and mapped to a feature vector in a TF.IDF fashion with respect to the set of positive and negative candidate tables in the example set.

The *layout features* are based on counts of rows, columns, cells (leaf text nodes) and the lengths of the cells. Here, rows refer to nodes declared as table rows by HTML-tags. We have not yet understood to what columns refer to in cases where the rows differ in the number of entries. The standard deviation of the number of cells per row is one of the layout features. All normalization are *local* in the sense that they refer to corresponding elements inside the same table.

Another layout feature is called *cumulative length consistency*. It is intended to capture the observation that the lengths of entries for a given column are often similar. The *content-features* transfer this concept to consistency with respect to content types such as *hyperlink, image, alphabetical*. This concept of consistency has also been the initial motivation for our work. While in our work this idea is formalized in terms of pairwise similarities, the consistencies in [14] are computed based on a reference value that is the mean for the lengths and the most frequent value for types. It might be interesting to draw a clear sight on the relation of these two formalizations, but we have not done this.

2.2 Display Models

In order to ease the reading of web pages, the developers of browser invest much effort to translate HTML-files into a clear arrangement one a 2-dimensional display. In [3], see also [4], the authors propose to make use of this arrangement result as provided by the mozilla rendering engine for solving the task of table extraction. We estimate this as a particularly clever approach. Tables can be characterized as boxes that are tiled in a regular manner fulfilling certain constraints. Though the arrangement pattern are of comprehensible complexity the approach yields good results on a large set of web-pages. It works out without any training.

The approach of [4] and the approach presented here are complementary in sense that the first focuses on topological structure of the input without considering the content of the table while the latter inspects the possible entries and separators without explicitly modeling the meaning of the separators for the arrangement. Nonetheless, they also overlap to some extend. Certainly, string similarity of separators is correlated to similar local topology. Also, in [4] congruence in text and background colors is taken into account which has no influence on the topology but on the similarity measures applied here.

2.3 Belief Propagation on Line Sequences

Pinto et. al. consider the problem of table extraction as segmentation of the input string into segments carrying labels such as *table-title lines, data-row lines, non-table lines* [10]. The authors target plain input strings and apply the assumption that segment borders are at line breaks such that entire lines can be used as tokens for linear belief propagation. Under a conditional random field model (CRF) this yields good results on set of documents fulfilling the assumption even with a rather simples feature set. In the provided state, the approach targets the location of the tables and meta-information on its rows but does not reveal the row-column structure of a table. To fix that the authors proposed to apply a 2-dimensional CRF with characters as token. For such a model, however, the optimization is difficult [13].

2.4 What is not Discussed

In this work we do not aim to extract meta-data as table titles or headers ([10]) nor provide additional tags carrying information that could used for further processing of the extracted data. The problem of integration of the extracted data [11] is not considered either, though this work is much inspired by [5] where data integration is a main application.

3 Notation and Assumptions

In this section we want get a more formal grip on what the candidate outputs of table extraction are. First, we formally specify the output space independent of concrete ground truths and extractors. Then, we formulate assumption made by the table extractor given in section 4. These assumptions only hold for less than one out of six input-output examples from the ground truth we build in section 6 but they allow us to work with a reduced output space that is suited for studying table extraction by column-wise similarities with a lean technical outlay.

3.1 Output Formalization

For table extraction we want to map an *input string* x over an alphabet Σ to a *list of tables* $y = (t^1, \dots, t^q)$. The k-th *table* in that list is a list of rows, $t^k = (r^1, \dots, r^{n_k})$, where the i-th *row* in turn consists of $m_{k,i}$ entries, $r^i = (e^1, \dots, e^{m_{k,i}})$. An *entry* is a string over Σ. If all rows of table t_k have an identical number of entries, m_k, the table is *rectangular*. In this case we can define the *columns* $(c^1, \dots, c^{m_k})$ of t_k as $c^j = (e^j_1, \dots, e^j_{n_k})$ such that $r^i = (e^1_i, \dots, e^{m_k}_i)$.

In contrast to output representation where entries are given as slices on the inpunt string, for instance [7], the above representation is *decoupled* in the sense that it does not tell us from where in the input the entries has been extracted, tough on real world data one usualy can reconstruct the entry positions. The decoupled output is more robust against input preprocessing and also more readable to human eyes. However, it demands a loss function that does not refer to the positions of the extracted substrings. The loss given in section **??** solves this by means of a best match evaluation.

3.2 Alternating Segmentations

In the following we formulate strong assumptions on the input string and the tables therein. These assumptions characterize the cases to which the proposed approach presented in this paper is restricted to.

The basic assumption is that the input string can unambiguously be segmented into an *alternating segmentation* of *tag segments* and *content segments*. The content segments are potential entries while a group of subsequent tags form potential separators between entries. More precisely, we assume that table entries always contain at least one non-whitespace character and the markups are given as non-overlapping *tags*, substring that start with < and end with > but have neither of both in-between. This way we can define a tag segment as sequences of tags only separated by whitespace. The content segments are segment between the tag segments.

A major demerit in these assumptions is that table entries can contain tags. But mostly they have formating purpose and surround a single content segment. Let the *peeling function* $\gamma : \Sigma^* \to \Sigma^*$ take away any prefix and suffix of its string argument that consist of tags and whitespace only. We assume that any table entry, when peeled by γ contains not further tags. Of course, there are entries that do contain tags surrounded non-tag, non-whitespace substrings. In such cases the assumptions do not hold and the algorithm below will fail.

The alternating segmentation is denoted by G and the separated subsequences of G containing separator segments and entry segments only by G^{s} and G^{e} respectively.

$$G = (g^{\mathrm{s}}_1, g^{\mathrm{e}}_1, g^{\mathrm{s}}_2, g^{\mathrm{e}}_2, \ldots, g^{\mathrm{s}}_p, g^{\mathrm{e}}_p),$$
$$G^{\mathrm{s}} = (g^{\mathrm{s}}_1, \ldots, g^{\mathrm{s}}_p),\ G^{\mathrm{e}} = (g^{\mathrm{s}}_1, \ldots, g^{\mathrm{s}}_p) \quad (1)$$

If the assumption does hold, the extraction of the table entries reduces to the selection of a subsequence of G^{e}. We further restrict the output space by additionally assuming that all tables rectangular and that they consist of consecutive content segments. This implies that there are no separating rows, which does not hold for a rather big fraction of tables. But this additional assumption reduces the space of candidate tables to a level that permits exhaustive search.

3.3 Reduced Search Space

Applying the above assumption, we can specify the space of candidate outputs for given input x in very simple terms. If the alternating segmentation of x has length p, any candidate table within x is can be represented as a triple (a, m, n) where a is the index of its first entry in G^{e}, m the number of columns, and n the number of rows such that $a + mn - 1 \leq p$. Let us denote the set such triples that represent one candidate table each by $T(G)$.

How many tables does $T(G)$ contain? Let $\mathrm{n}^{\mathrm{p}}(a)$ be the number of tables with start index a, and $\mathrm{n}^{\mathrm{c}}(l)$ the number of tables that can be built from at most l consecutive content segments in G^{e}. Since $\mathrm{n}^{\mathrm{p}}(a) = \mathrm{n}^{\mathrm{c}}(p - a + 1)$, we have $|T(G)| = \sum_{a=1}^{p} \mathrm{n}^{\mathrm{p}}(a) = \sum_{l=1}^{p} \mathrm{n}^{\mathrm{c}}(l)$. The addend $\mathrm{n}^{\mathrm{c}}(l)$ equals the number of ordered pairs that multiply to at most l and hold the following bounds for all $l > 0$.

$$l(\ln l - 1) - 2\sqrt{l} \leq \mathrm{n}^{\mathrm{c}}(l) \leq l(\ln l + 2) + 2\sqrt{l} \quad (2)$$

The number of candidate tables is therefore bounded from above by $p^2(\ln p + 1)$.

4 Extraction by Column-Wise Similarities

We want to study a simple score based table extraction algorithm. The algorithm is named *CSE*, standing for *column-wise similarity evaluation*, as its selects the output tables according to a table score based on the in-column similarities.

4.1 Iterative Maximization

The proposed table extraction algorithm tries to solve the table location and recognition task in one go by maximizing a *table score* H over the candidate table in $T(G)$ which will be given below. The maximizer $\hat{t}$ of H is one table in the output list $\hat{y}$.

$$\hat{t} = \underset{(a,m,n) \in T(G)}{\operatorname{argmax}} H(a, m, n) \quad (3)$$

The other tables in $\hat{y}$ are obtained by incrementally excluding candidate tables that overlap with the tables extracted so far and select the maximizer out of the remaining ones. The output list is finally sorted by the position of the first entry a such that the tables appear in the same order as they do in the input string.

We let aside the problem of detecting where in the sequence of maximizers the entries stop to be tables. That is, the algorithm gets along with the input string the true number of tables q as *promise*, using a notion from complexity theory [1]. In case that the first $l < q$ tables already cover the sequence such that no further table can be extracted, the remaining tables are defined as empty tables that are tables with zero rows.

4.2 Table Score

The table score should express how table-like a table is. It is exclusively obtained from scores on the columns of the table, where column means both, column of entries and column of separator between entries and rows. Row separators are treaded like entry separators. The difference is that in a table there is one row separator less than there are separators between two content columns because we model a table to start with the first entry of its first row and to end with the last entry of its last row.

A table column is represented by a tuple (u, b, m, n) where $u \in \{\mathrm{e}, \mathrm{s}\}$ is its type (entry- or separator-column), b is the index of the first entry of the column in G^u, m is the number of columns the table to that the column belongs to has and n is the number of element in the column. This means that the quadruple refers to the column that consists of the segments $g^u_b, g^u_{b+m}, g^u_{b+2m}, \ldots, g^u_{b+(n-1)m}$. We denote the score of that column by $h^u(b, m, n)$.

For a given table $(a, m, n) \in T(G)$ the table score is the sum of scores of its columns divided by a normalization term $z(m,n)$.

$$H(a,m,n) = \frac{1}{z(m,n)}(h^{\mathrm{e}}(a,m,n) + \sum_{j=1}^{m-1}(h^{\mathrm{s}}(a+j,m,n) + h^{\mathrm{e}}(a+j,m,n)) + h^{\mathrm{s}}(a+m,m,n-1)) \quad (4)$$

The first addend refers to the first entry column, the last addend is the score of the separator column that contains the row separators. The latter has one element less than the other columns of the table. Also note that according to (1) the separator segment with index a comes before the content segment with the same index.

For either type u we aim the score of a column (u,b,m,n) to indicate how well the elements of the column $(g_{a+im})_{i=0,\dots n-1}$ in G^{u} fit together. We can model this by the sum of their pairwise similarities. Let $\bar{s}^u : Q^u \times Q^u \rightarrow [0,1]$ be a similarity measure where Q^u is the set of possible segments of type u. Then, the score of a column (u,b,m,n) is given by

$$h^u(b,m,n) = \sum_{0\le i<j<n} \bar{s}^u(g_{b+im}, g_{b+jm}). \quad (5)$$

The normalize term is the total number of similarities that are taken into account

$$z(m,n) = (2m-1)\binom{n}{2} + \binom{n-1}{2} \quad (6)$$

such that H is the average similarity between entries or separators stemming from the same column.

4.3 Entry Similarities

A good design of the similarity functions s^{e} and s^{s} presumably is an important factor of the performance of the CSE extraction algorithm. However, we did not undertake systematic studies on what may be an adequate choice of similarities nor run experiments in order to compare their effectiveness. In the following we briefly describe the similarities we applied in our experiments presented in section 6.

For both segment types with apply a similarity s^{l} based on the segment lengths. Let us write $|x|$ for the length of a string x. The length similarity of two segments a and b evaluates the ratio of the greater length to the smaller one through an exponential decay.

$$s^{\mathrm{l}}(a,b) = \exp(-g(a,b)), \quad g(a,b) = \frac{1+\max(|a|,|b|)}{1+\min(|a|,|b|)} - 1 \quad (7)$$

While the similarity of separator segments is reduced to the length similarity, e.g. $s^{\mathrm{s}} = s^{\mathrm{l}}$, the similarity on entry segments additionally checks whether the two string are of the some type where the type of string is either *integer*, *non-integer number*, or *other*. This type similarity $s^{\mathrm{t}}(a,b)$ is 1 if the types of a and b match, and 0 otherwise. The entry similarity is given as product of length and type similarity: $s^{\mathrm{e}}(a,b) = s^{\mathrm{t}}(a,b)s^{\mathrm{l}}(a,b)$.

In principle we one can plug-in any string kernel as similarity function on segments. Unbounded similarities have to be used trough normalization, $\bar{s}^u(g,\bar{g}) = s^u(g,\bar{g})/\sqrt{s^u(g,g)s^u(\bar{g},\bar{g})}$. It should be noted that the evaluation of the similarities must have moderate costs because any two segments of the same type are to be compared.

5 Score Levelling

In the previous section we proposed to take averaged similarities of table component as selection criterion for tables. Unfortunately, this does not work out because of two reasons. To the first we refer as *subtable problem*. Consider two tables, where one table is a subtable of the other, both having approximately the same elevated score. In such a case we prefer the greater table, because we assume that a wrong extension of a complete table decreases the score while false dropping of rows does not so.

The second issue is the problem of *maximization bias*. For smaller shapes, we average over fewer similarities, tables have less average overlap to each other, there are more non-overlapping candidates in absolute numbers. These properties of the score make the selection by score maximization favor smaller shapes, even if the expected score does not depend on the shape.

The two issues differ in their scope. The subtable problem refers to preferences among certain outputs having similar score H. In other settings we might use a similar score although no or other preferences exists. In contrast, the maximization bias is not related to the input-output distribution but refers to the different predictive qualities of the scores due to the score structure.

5.1 Linear Levelling

We tackle these issues by means of *score levellings* that map the score H of a candidate table to a leveled score $\bar{H}$. The levellings are *increasing in the shape* that is the original score is decreased the more the smaller m and n are. We confine ourself to *linear levellings* that have the form below. For better reading, a single term s is used to denote the shape (m,n).

$$\bar{H}_G(a,s) = \frac{H_G(a,s) + b_G(s)}{c_G(s)} \quad (8)$$

A pragmatic approach would be to try a reasonable guess for c and b and use it as *ad hoc levelling* or to fit the levelling from a broader class of functions using a training set. Instead, we discuss in the subsequent subsections ways to tackle the maximization bias by levellings of the form (8) explicitly. In contrast, we do not respond to the subtable problem directly. We assume that all levellings that do not increase to slowly will sufficiently solve it. If the subtable is much smaller, the score of the supertable will be decreased much less. If, on the other hand, the subtable only misses a few rows, then its score is unlikely to be much greater since it contributes the main part to the score of the supertable.

In the remainder of this subsection we try to give a better idea of the levelling approach and therefore use simplified setting: we assume the input-output pairs $Z = (G,K)$ are drawn from a distribution P which only supports segmentations G that have a given length p and contain exactly one true table denoted by K. We say P has *input length* p and *output length* 1.

Let l be a *score loss* on candidate tables, for instance the 01-loss $l_Z(t) = [\![H_G(t) > H_G(K)]\!]$ or the *hinge-loss* $l_Z(t) = \max(0, H_G(t) - H_G(K))$. The *risk of the score* $\mathrm{R}_P(H)$ is the expected sum of losses over all candidate table in $T(G)$ that is $\mathrm{R}_P(H) = \mathrm{E}_{Z\sim P}\sum_{t\in T(G)} l_Z(t)$, where $Z \sim P$ says that Z is drawn from P. We want to decompose that risk along the shapes. Let S_p be the set of *feasible shapes*, $A_p(s)$ be the set of *feasible positions* given the shape s, and $e(s) = \mathrm{E}_{G\sim P}(e_G(s))$ with

$e_G(s) = \sum_{a \in A_p(s)} l_Z((a,s))$ be the *risk at shape* s.

$$\mathrm{R}_P(H) = \sum_{s \in S_p} e(s). \tag{9}$$

The approach of shape levelling assumes that independently of P but due to the structure of H the risk e is greater for certain shapes such that reducing the chances of a such an s reduces $e(s)$ more than it increases the risk at other shapes and therefore leads to a smaller total risk which in turn is assumed to correspond to a better extraction performance. Note that we do not further discuss that notion of risk nor the choice of the underlying score loss. Here, they are only used in order to explain the concept of levelling but they are not directly taken into account when we discuss ways to define levellings below.

5.2 Fair Levelling

The idea of *fair levelling* is to design the levelling such that the score maximization scheme does not favor any shapes when the segmentation can be assumed to contain no tables.

Let P be of input length p and output length 0. The segmentations drawn according to P do not contain any tables and we therefor call to such a P *table-less*. Given a table-less distribution P, we say that the table score is *fair with respect to* P if the shape s^* of the maximizer of H_G has uniform distribution over S_p.

A sufficient condition for obtaining a fair score is that that distribution of the maximum does not dependent on the shape. As this goal is overstated we propose a rough approximation thereof. Let $\mu_p = \mathrm{E}_{G \sim P} \mu_G$ be the expected average score over S_p and $A_p(s)$ and let $\nu_p(s) = \mathrm{E}_{G \sim P} \nu_G(s)$ be the expected maximum of the scores over $A_p(s)$.

$$\mu_G = \operatorname*{avg}_{s \in S_p, a \in A_p(s)} H_G(a,s) \quad \nu_G(s) = \max_{a \in A_p(s)} H_G(a,s) \tag{10}$$

With the following levelling, we approximate a fair levelling by standardizing the expected maxima given the shape, e.g. we set $\mathrm{E}_{G \sim P} \max_a \bar{H}_G(a,s) = 1$ for all s in S_p.

$$\bar{H}_G(a,s) = \frac{H_G(a,s) - \mu_G}{\nu_p(s) - \mu_p} \tag{11}$$

In praxis, we have to use estimations of μ_p, $\nu_p(s)$ that we obtain in our experiments in section 6 simply by averaging $\nu_G(s)$ and μ_G over a set of segmentations drawn from some P.

5.3 Table-Less Models

The approach of approximated fair levelling demands that we have a table-less distribution P at hand. We discuss three simple *table-less models*.

The first model, called *Bernoulli model*, is a simply *iid model* where we draw the segments independently and with equal chances from $\{0,1\}$. The similarity of two segments is 1 if they are equal and 0 otherwise. This model has little to do with the segmentation from which we want to extract tables but still might be sufficient to design a effective levelling as it does capture the structure of the table scores.

The second model, which is named *shuffling model*, is an iid model as well. A segmentation is drawn from an example set and then we sample the segments for the new segmentation according to the distribution of segments in the sampled segmentation. At least with high probability we can assume that we do not find any table in a segmentation that is drawn according to either of these iid models.

Last, we consider the *empirical model* where we sample a segmentation by randomly drawing it from a set of example segmentations that containing no tables. From one segmentation of length p we obtain sample value of $\nu_p(s)$ for any $s \in S_p$. But contrary to the iid models, we only get empirical evidence for some p and therefor need a more elaborate smoothing technique than for the iid models where we can generate segmentation for any p. On the other hand, we assume the levelling to be monotone in each of its the arguments m, n and p what strongly decreases the data demand. Nonetheless, the definition of such a smoothing remains future work.

The empirical model as as well as the shuffling model amount to supervised table extraction since we only want to sample segmentation not containing tables. Though, that binary labeling is rather cheap compared to the extraction of the tables.

5.4 Variance Levelling

We conclude this section we simple levelling scheme, called *variance levelling*, where we standardize the score in the classical sense with respect to some table-less distribution P. That is, we divide the score by the standard deviation that the tables of shape s are expected to have when we run over the feasible positions.

$$c(s)^2 = \mathrm{E}_{G \sim P} \operatorname*{avg}_{a} (H_G(a,s) - \mu_G)^2 \tag{12}$$

With an iid model we can explicitly compute the values $c(s)$ from two parameters of the underlying segment distribution. For simplicity, we now include the separator subsequent to the last entry segment into the table such the table score is the sum of $2m$ independent identically distributed columns scores. Let H be the score of some candidate table in $G \sim P$ with shape (m,n) and let C be the score of a column in G having n entries. Taking the column score as U-statistic for a binary kernel, we have

$$\begin{aligned} \mathrm{V}(H) &= \frac{1}{2m}\mathrm{V}(C) && (13) \\ &= \frac{1}{mn(n-1)}\left((n-2)\sigma_1^2 + \sigma_2^2\right) && (14) \end{aligned}$$

where $\sigma_1^2 = \mathrm{V}_X(\mathrm{E}_Y(s(X,Y)))$ and $\sigma_2^2 = \mathrm{V}_{X,Y}(s(X,Y))$, see for instance [8]. For the Bernoulli model the parameters σ_2^1 and σ_2^2 can easily obtained from success probability q. In case of the shuffled model we have to estimate them by sampling.

This standardization is simpler than the approximated fair levelling approach but the motivation is less sound. One the one hand, high variance certainly is an important factor for the maximization bias problem. On the other hand, the distribution of the maximum is not determined by the mean value and the variance.

6 Experiments

For testing the general performance of the CSE algorithm and for comparing the different levellings presented above we run experiments on the Wang and Hu data set that was used in [14].

6.1 Building the Test Set

The Wang and Hu set consists of HTML-documents in which all target tables are marked using a special boolean attribute in the table node. This makes the set a ready-to-use ground truth for the table location task. Since CSE tries to solve the table recognition task as well, we have to extend the ground truth provided by Wang and Hu by additionally solving the recognition of table cores.

We decided to do this automatically with another table extractor, named *RE extractor* or *REE* in short, that uses regular expressions based on the relevant element names of HTML. Our attempts to parse the document tree failed for too many documents where we tried different freely available parsers including the lxml-library [9].

REE uses a pattern for target tables based on the element name *table* and the additional attribute in order to get the substrings that represent the content of a target table. To solve the recognition task it applies an entry pattern (element name *td*) on the content of the matches of the row pattern (element name *tr*), which in turn is applied on the substrings matching the target table pattern. Matches of the row pattern that contain matches for headers (element name *th*) are ignored.

REEs capability for solving the table recognition task is limited. One minor problem is broken HTML in the inspected substrings. The main issue is meta data that is not declared as such. Still, we belief that the output of REE is sufficient for our experiments since extraction capability of CSE is rather low in any case.

In the following we specify two versions of the ground truth. Both are based on the output of REE but they apply filters of different strength. While for the *full set* only weak filtering is applied, *feasible set* contains those cases only that fulfill the assumption made by CSE.

Full Set

The Wang and Hu data set contains a total of 1393 documents. For the *full set* we only include the examples that contain at least one target table. CSE gets the number of tables k as promise and therefor has nothing to do if the promise is 0. Further, we bound the segmentation length to be not greater than 900. Documents with $p > 900$ are rare but they take a rather big fraction of the total runtime.

As a third filter criterion we demand that the extraction by REE is *successful* in the following sense: each extracted table should contain at least one row and any extracted row should have at least one entry. We hope that many cases where the table recognition by REE does not work as intended are detected by this criterion, while not to many table that are extracted as intended by REE fail to fulfill it. Table 1 shows the number of examples passing the filters discussed so far plus an additional filter discussed below.

Feasible Set

The feasible set is restricted to those documents from the full set in which all tables provided by the RE extractor fulfill the assumptions given in section 3.2. Though CSE may extract some of the table or part of tables from a document not in the feasible set, it is not possible that its output is entirely correct. The feasible is useful to analyze the discriminative power of in-column similarities as used by CSE and variants of the algorithm.

In order to fulfill assumptions a table has to be rectangular and it has to *fit in the content sequence* of the document. The latter means that the sequence of the entries in the table is a consecutive subsequence of the content part G^{e} of the alternating segmentation modulo the mapping γ. The conjunction of this two criteria is necessary for the assumptions to hold, but unfortunately it is not sufficient because of the occurence implicit headers. However, this problem can only be solved by human inspection and we therefor prefer to the use the above criteria as approximation. The number of documents under *hold assumptions* in table 1 refers to this approximation.

The fraction of documents from which REE erroneously extract meta data presumably is lower in the feasible set because the rectangularity condition filters out cases with implicit titles or separating rows that are given as rows with one or no entry.

6.2 Performance Measure

For the evaluations of an extractors we need a loss function L^{y} that compares its outputs to the output provided as ground truth and encodes the comparison as a value in $[0, 1]$. The definition of the loss goes along the hierarchical structure of the outputs them self: L^{y} is an extension of a loss on tables L^{t} that is an extension of a loss on rows L^{r} which in turn is an extension of a loss on entries L^{e}.

We say that an extension is *strict* if the resulting loss is 1 whenever the two arguments do not have the same number of components and otherwise is given as aggregation of the *component losses* that are the losses on the pairs of components one from each of the two arguments having an identical index. The *average extension* is a strict extension which aggregate by the mean and the also strict *maximum extension* takes the maximum as aggregation. For instance, we obtain the 01 *loss* that checks whether its arguments are equal down to their entries by applying the maximum extension at any level of the hierarchy. Here, we want use a loss on table list which is more soft to several regards as we define in the following in a bottom-up manner.

The loss L^{e} on two entries is the 01 loss L^{s} on strings applied to the entries reduced by the peeling function γ introduced in section 3.2.

$$L^{\mathrm{e}}\left(e, \bar{e}\right) = L^{\mathrm{s}}\left(\gamma(e), \gamma(\bar{e})\right) \tag{15}$$

While the row loss L^{r} is given as strict maximum extension of the loss on entries, we want to use a soft table loss L^{t} such that dropping rows at the beginning or the end of a table results only in a gradual increase of loss.

Therefor, we define the table loss L^{t} not as a strict extension but as *best overlap extension* of the row loss. This extension searches an optimal way to match consecutive subsequence of component indexes of the argument with the smaller number of components to the longer one. For every component of the longer argument that is not matched a loss of 1 is taken into account.

Let $t = (r^1, \ldots, r^n)$ and $\bar{t} = (\bar{r}^1, \ldots, \bar{r}^{\bar{n}})$ be two tables that are to be compared where we assume without loss of generality that $n \leq \bar{n}$. In order to simplify the below definition of the loss L^{t} on the tables, we extend the shorter table t to $\tilde{t} = (\tilde{r}^1, \ldots, \tilde{r}^{\bar{n}+n+\bar{n}})$ by adding $\bar{n}$ *false rows* to either end of t. A false row is a row r such that $L^{\mathrm{r}}(r, \bar{r}) = 1$ for any row $\bar{r}$.

$$L^{\mathrm{t}}\left(t, \bar{t}\right) = \min_{d=0,\ldots,n+\bar{n}} \frac{1}{\bar{n}} \sum_{i=1}^{\bar{n}} L^{\mathrm{r}}\left(\tilde{r}^{d+i}, \bar{r}^i\right). \tag{16}$$

The minimization is needed because we want to define a loss depending only on the outputs decoupled from the input as pointed out in 3.1. If we toke the entry slices on the input string into account, we could use them to match the rows directly.

Finally, the table loss is expanded to a loss on table lists L^{y} by applying the average extension. The strictness of the

total	$k > 0$	$p \leq 900$	RE successful	hold assumptions
1393	774	727	700	162

Table 1: The number of examples in the Wang and Hu set that passed the filters in conjunction from the left to right up to the filter heading the column.

extension is not an issue here because the CSE extractor uses promises on the number of tables. We refer to output loss L^{y} as *best row overlap loss* or *BRO loss* in short.

6.3 Results

We evaluated the performance of the CSE extractor by comparing its output to the REE-ground-truth discussed in subsection 6.1 in terms of best row overlap loss defined subsection 6.2. The CSE extractor gets the number of tables obtained from REE as promise. To simplify the implementation we let CSE look only for tables with $n \leq 80$, $m \leq 20$, other candidate tables were ignored.

In section 5 we pointed out that one has to adjust the scores depending on the shape of the candidate table in order to make the extraction by average entry similarities work. The discussed levellings try to do this adjustment in a specific, roughly motivated way. Alternatively, one may pass on the motivation and try to make a good guess on a function of m and n and use this as *ad hoc levelling*. For instance, one might multiply the scores by

$$\frac{1}{c(m,n)} = \gamma + n^{\beta} m^{\alpha\beta} \tag{17}$$

for some α, β and γ. Of course, we do not know a priory how to these set parameters. One can fit them using a training set, but we not try this possibility for this work. Still, we used the above class of ad hoc levellings for two purposes. First, by varying the parameters we get a rough impression on the impact of changes in the levelling. Second, we consider it as base line in the sense that the results yielded by levelling from section 5 should be comparable to this ad hoc levelling at least if the chosen parameters had not seriously been fitted to the test set. As a matter of fact, it was not difficult to find parameters for the ad hoc levelling in (17) that give better result than the more elaborated levellings discussed in section 5.

In general, the extraction performance of CSE with features from section is rather poor: the BRO losses are 0.825 and 0.613 for the full set and feasible set respectively using ad hoc levelling with $\alpha = \beta = \gamma = 0.5$. Results on the feasible set for fair and variance levelling based on Bernoulli sampling with different success probabilities q, for fair levelling with shuffled sampling, as well as for one ad hoc levelling are given in table 2.

The Parameters for the levellings in table 2 were chosen as follows. For the ad hoc levelling the result refers to the optimal values where we tested all combination with $\alpha = 0.2, 0.4, 0.6, 0, 08$, $\beta = 0.2, 0.3, \ldots, 1.4$ and $\gamma = 0, 1, 2, 8, 32$. The success probability for the iid Bernoulli segment sampling was tested at $q = 0.1, 0, 2, 0.3, 0.4, 0.5$ and the values that yielded the best and the worst performance respectively are in given in the table. Therefor, neither of those performances can be stated as performance for respective type of levelling as the parameters are fitted to the test set. The fair levelling with shuffled sampling is based on samples that are also taken from the Wang and Hu set but do not belong to the test set as they do not contain tables.

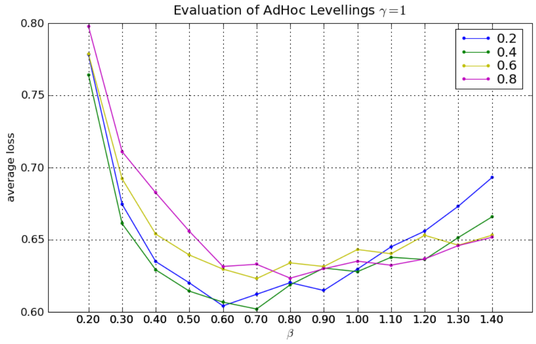


Figure 1: The mean BRO loss obtained by CSE with ad hoc levelling for different values of α and β measured on the feasible set. Different values of α are plotted with different colors while β runs along the x-axis.

Figure 1 shows the BRO losses for ad hoc levelling with $\gamma = 1$. The corresponding plots for other values of γ have similar shapes where with increasing γ the low loss region shifts towards greater values of β and the graphs for different α approaches to each other.

7 Conclusion

We investigated in a simple approach to the task of table extraction: first, we reduce the output space such that we can afford to inspect any candidate output, then, we select a given number of candidates with high average in-column similarities. The inspection of a set of HTML documents revealed that the proposed reduction of the output space cannot be applied in too many cases. Experiments on the remaining cases gave a first impression on the discriminative power of in-column similarities: even with more elaborated entry similarities than the simple ones applied here, it is presumably to weak to sufficiently solve the extraction task. On the other hand, given the simplicity of the applied similarities, we find that the extraction performance is on a level that justifies deeper investigation how we can effectively use the information that lies in similarities of column entries. In the following we revisit some issues of the this approach and indicate proposals for future work.

7.1 Alternating Segmentations

In section 3.2 we defined the input segmentation in terms of SGML tags but the concept of alternating segmentation is more general. Instead of tag-segments vs non-tag-segments on texts with markups, one might consider other input types with other alternating segmentations. A sufficient condition for an alternating segmentation in general terms is that no suffix of a separator-segment is a prefix of a content-segment *or* vice versa. Further, we can build different alternating segmentations and jointly maximize over the contained candidate tables, provided that the scores functions yield comparable values.

Ad Hoc	Fair Bern. $q = 0.5$	Fair Bern. $q = 0.2$	Fair Shuffled	Var. Bern. $q = 0.1$	Var. Bern. $q = 0.4$
0.598	0.655	0.617	0.623	0.664	0.628

Table 2: BRO losses of CSE using different type of levelling measured on the feasible set. The parameters for the ad hoc levelling in the first column are $\alpha = 0.4, \beta = 0.6$ and $\gamma = 0$ yielding the lowest among all tested combinations. For fair and variance levelling the two given values of q yielded the worst and the best result among five tested values from 0.1 to 0.5.

7.2 Restrictive Assumptions

The assumptions formulated in section 3.2 are too restrictive. Only one out four documents from the full set does fulfill them. Partially this caused by meta data rows in the ground truth provided by REE but we believe that that fraction would not increase to reasonable level even if we cleaned the example output by hand. It should be noted that most relaxations cause a exponential blowup of the search space. For instance, if we allow table rows to be separated by one or two separator segments instead of exactly one, the number of candidate table starting at given position grows exponentially in the number of rows as long as there are segments left in the segmentation. It is not obvious how to solve the maximization efficiently under such a relaxation. We cannot apply standard dynamic programming along the sequence of segments, because the column scores as given in section 4 does not factorize along this sequence.

7.3 Evaluation of Levellings

Except for levelling with shuffled sampling, all levellings that have been applied in our experiments are parametrized. As long as we do not provide algorithms for pre-test determination of the parameters a comparison of levellings schemes based on the obtained performances is delicate. But we might say that fair and variance levelling as proposed in section 5 do not provide a adequate technique for boosting the performance of CSE compared to ad hoc levelling since competitive parameters can easily be found for the latter. The proximate way to make comparison between parametrized levellings is to fit each of the levellings with respect to a training set. This will also give us an idea to what extend the performances differ on disjoint test sets which is important to know when we decide on the effort to put in the selection of the levelling.

7.4 Training of Similarity Functions

The definition of the kernels in sections 4.3 were made ad hoc but we belief that more elaborated similarities can improve the performance of the CSE algorithm. In particular, we would like to adapt the similarity functions to the extraction task based on training examples. The table score given in section 4.2 is suitable for training. At least for a given levelling scheme, the score of a candidate output is linear in any linear parameterization of the similarities, for instance linear combinations on a set of fixed kernels. Provided such a parametrization, we can apply generic training schemes for linear models as the one proposed in [12]. However, for sake of linearity, we have to restrict ourself to documents that contain one table only. Further, we depend on a fast convergence since the evaluation of H is expensive.

References

[1] J. L. Balcazar, J. Diaz, and J. Gabarro. *Structural Complexity I.* EATCS Monographs on Theoretical Computer Science. Springer Verlag, 1993.

[2] H.-H. Chen, S.-C. Tsai, and J.-H. Tsai. Mining tables from large scale HTML texts. In *International Conference on Computational Linguistics (COLING)*, pages 166–172. Morgan Kaufmann, 2000.

[3] W. Gatterbauer and P. Bohunsky. Table extraction using spatial reasoning on the CSS2 visual box model. In *National Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2006.

[4] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krüpl, and B. Pollak. Towards domain-independent information extraction from web tables. In C. L. Williamson, M. E. Zurko, P. F. Patel-Schneider, and P. J. Shenoy, editors, *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 71–80. ACM, 2007.

[5] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Neural Information Processing Systems*, pages 513–520. MIT Press, 2006.

[6] M. Hurst. Layout and language: Challenges for table understanding on the web. In *In Web Document Analysis, Proceedings of the 1st International Workshop on Web Document Analysis*, pages 27–30, 2001.

[7] N. Kushmerick. Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*, 118(1–2):15–68, 2000.

[8] A. Lee. *U-Statistics: Theory and Applications*. Marcel Dekker Inc., New York, 1990.

[9] lxml: pythonic binding for the libxml2 and libxslt libraries. http://codespeak.net/lxml/index.html.

[10] D. Pinto, A. McCallum, X. Wei, and W. B. Croft. Table extraction using conditional random fields. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information*. ACM, 2003.

[11] A. Pivk. Automatic ontology generation from web tabular structures. *AI Communications*, 19(1):83–85, 2006.

[12] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 104, New York, NY, USA, 2004. ACM Press.

[13] M. J. Wainwright and M. I. Jordan. Semidefinite relaxations for approximate inference on graphs with cycles. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Neural Information Processing Systems*. MIT Press, 2004. (long version).

[14] Y. Wang and J. Hu. A machine learning based approach for table detection on the web. In *Proceedings of the Eleventh International World Wide Web Conference*, pages 242–250, 2002.

Combinations of Content Extraction Algorithms

Yves Weissig[1], Thomas Gottron[2]
[1]Technische Universität Darmstadt, 64289 Darmstadt, Germany
[2]Johannes Gutenberg-Universität Mainz, 55099 Mainz, Germany
weissig@rbg.informatik.tu-darmstadt.de, gottron@uni-mainz.de

Abstract

Content Extraction is the task to identify the main text content in web documents – a topic of interest in the fields of information retrieval, web mining and content analysis. We implemented an application framework to combine different algorithms in order to improve the overall extraction performance. In this paper we present details of the framework and provide some first experimental results.

1 Introduction

HTML documents on the web are composed of far more data than their main content. Navigation menus, advertisements, functional or design elements are typical examples of additional contents which extend, enrich or simply come along with the main content. This “noise” in the documents contributes for about 40 to 50% of the data on the World Wide Web [Gibson *et al.*, 2005].

Cleaning web documents from this additional contents improves performance of information retrieval, web mining and content analysis applications. The task of identifying and/or extracting the main text content of a web document is most commonly referred to as Content Extraction (CE). Several good algorithms have been developed and evaluated in the last years. However, only the Crunch system [Gupta *et al.*, 2003] tried to approach the task with a fixed ensemble of different algorithms. As the heuristics employed in Crunch performed better in combination than when they are used individually, we motivated in [Gottron, 2008a] to look closer at the potential of using ensembles of CE algorithms. This lead to the idea of the proposed CombinE system as outlined in figure 1. Its aim is to flexibly incorporate the results of filter modules implementing different algorithms and, thereby, obtain better or more reliable extracts of the main content.

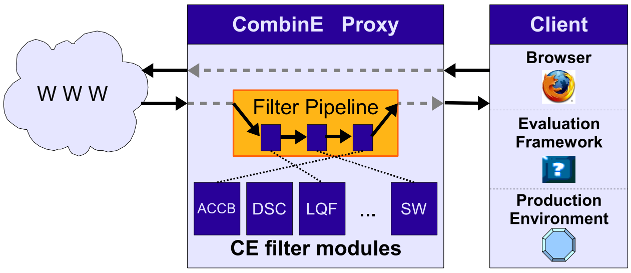


Figure 1: Outline of the *CombinE* system.

In this paper we show how combinations of CE algorithms can be realised and present some first results on how successful such combinations can be.

We proceed with a look at related work in the next section and a short description of our application in section 3. Then we focus on merging different document extracts in 4 and take a look at some first CE combinations and their performance in 5, before concluding the paper in 6 with a prospect at future work.

2 Related Work

The number of CE algorithms available for cleaning HTML documents is increasing. Especially in the last years the topic seems to receive more attention than before, probably due to the increased amount of noise in web documents.

The *Body Text Extraction* (BTE) algorithm [Finn *et al.*, 2001] interprets an HTML document as a sequence of word and tag tokens. It identifies a single, continuous region which contains most words while excluding most tags. A problem of BTE is its quadratic complexity and its restriction to discover only a single and continuous text passage as main content. Pinto et al. extended BTE in their *Document Slope Curves* (DSC) algorithm [Pinto *et al.*, 2002]. Using a windowing technique they are capable to locate also several document regions in which the word tokens are more frequent than tag tokens, while also reducing the complexity to linear runtime. *Link Quota Filters* (LQF) are a quite common heuristic for identifying link lists and navigation elements. The basic idea is to find DOM elements which consist mainly of text in hyperlink anchors. Mantratzis et al. presented a sophisticated LQF version in [Mantratzis *et al.*, 2005], while Prasad and Paepcke describe how to learn a weighting scheme for LQF in [Prasad and Paepcke, 2008]. *Content Code Blurring* (CCB) [Gottron, 2008b], instead, is based on finding regions in the source code character sequence which represent homogeneously formatted text. Its ACCB variation, which ignores format changes caused by hyperlinks, performed better than all previous CE heuristics. Weninger and Hsu proposed a line based approach to find text-rich areas in [Weninger and Hsu, 2008].

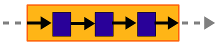
Evaluation of CE algorithms was addressed in [Gottron, 2007]. Providing a goldstandard for the main content, the idea is to compute the intersection of a CE algorithms’ output with the goldstandard using the longest common subsequence [Hirschberg, 1975] over the word sequences. This allows to apply classical IR measures like Recall, Precision and F_1. The fact, that such an evaluation can be run automatically and unsupervised is exploited in [Gottron, 2009] to construct a framework for parameter optimisation based on genetic algorithms.

3 The CombinE System

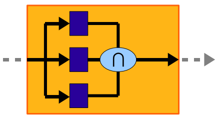
The aim to flexibly combine different CE algorithms in order to obtain better and/or more reliable results motivated the idea of the CombinE System [Gottron, 2008a]. It is an http-proxy server which can filter documents on-the-fly when receiving them from the web. An outline of the overall system was already given in figure 1.

To filter the web documents, CombinE uses an extensible set of CE algorithms with a standardised interface. The combination of algorithms is modelled in filter sets which can be realised in different ways:

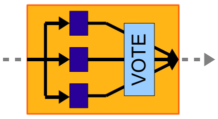
Serial The CE algorithms are applied in a predefined order, where the output document of one filter provides the input for the next one.



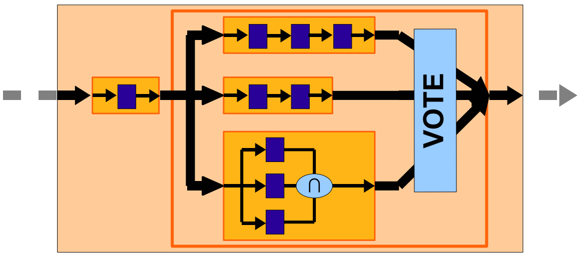
Parallel Several CE algorithms are applied to individual copies of the original document separately. The results are then combined by either unifying or intersecting the results. We get to the details of how to do this in the next section.



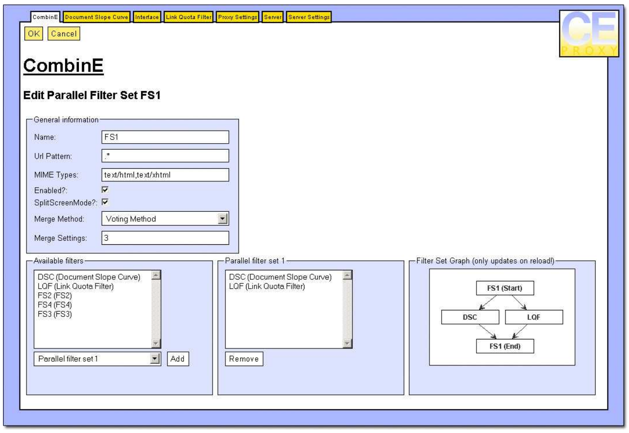
Voting A special case of the parallel combination is a voting setup. Each CE filter can vote for parts of a web document to belong to the main content. Only if a content fragment gets enough votes from the filters, it is finally declared to actually be main content. Attaching weights to the votes provides a further possibility to influence the results.



As technically each of this combinations of CE algorithms acts like a CE algorithm itself (HTML document as input and cleaned HTML as output), they can be incorporated in other filter pipelines. The following example demonstrates how filter sets can be combined into new, more complex sets.



The CombinE proxy can be configured comfortably over a web interface (c.f. figure 2). Users can choose from a selection of standalone CE algorithms as well as already combined versions to create new serial, parallel or voting filter sets.



CombinE

Edit Parallel Filter Set FS1

General information

Name: FS1

Url Pattern: .*

MIME Types: text/html,text/xhtml

Enabled?: ☑

SplitScreenMode?: ☑

Merge Method: Voting Method

Merge Settings: 3

Available filters

DSC (Document Slope Curve)
LQF (Link Quota Filter)
FS2 (FS2)
FS4 (FS4)
FS3 (FS3)

Parallel filter set 1

Parallel filter set 1

DSC (Document Slope Curve)
LQF (Link Quota Filter)

Filter Set Graph (only updates on reload!)

Figure 2: Web interface for creating and managing filter sets.

4 Merging Extraction Results

The implementation of the different merging methods used within CombinE was one of the most challenging tasks in the project. While the serial approach was easy to implement because the output of the previous filter was used as the input of the next filter, the parallel combination is a non-trivial issue. A filter set can have up to N different sub filters which need to be executed. After each of the N filters returned a result it is the task of the merging method to integrate the N results into one message that is returned to the user.

CombinE uses a merging method based on DOM-trees. The generation of a DOM-tree out of an http-message is obtained via the NekoHTML parser[1], which is flexible, fast and reliable. We also use NekoHTML to normalise the HTML documents into a form which is compliant with W3C specifications.

The overall process of parallel filtering can be subdivided into the following steps:

- Pre-process the http-message, normalise HTML.
- Filter copies of the pre-processed http-message with each filter. The result is an array containing N filtered http-messages.
- Convert each of the filtered messages into DOM-trees.
- Merge the N DOM-trees with the merging method chosen for the current filter pipeline. The result is a single DOM-tree.
- Transform the DOM-tree into an http-message.
- Return the resulting http-message.

We consider the DOM-tree as a tree with M nodes. The nodes represent different HTML-elements, e.g. a <table>-Tag, an <img>-Tag or a text-value. The intersection merging returns only elements appearing in each of the N filtered documents. The result can be described as $R_\cap := \{x \mid \forall i \in N : x \in D_i\}$ with x the nodes of the DOM-trees, so that R is the set of all elements contained in all D_i. In contrast to this, the union merging returns the elements that appear in any one of the filtered documents. It can be described as $R_\cup := \{x \mid \exists i \in N : x \in D_i\}$.

While the calculation of general tree mapping is a complex task, we can assume the CE filters to at most remove nodes from the DOM-tree. Hence, in an recursive approach, we start from the DOM-tree root nodes and collect all child nodes that are present in all (intersection) or any (union) of the DOM-trees.

[1] http://sourceforge.net/projects/nekohtml

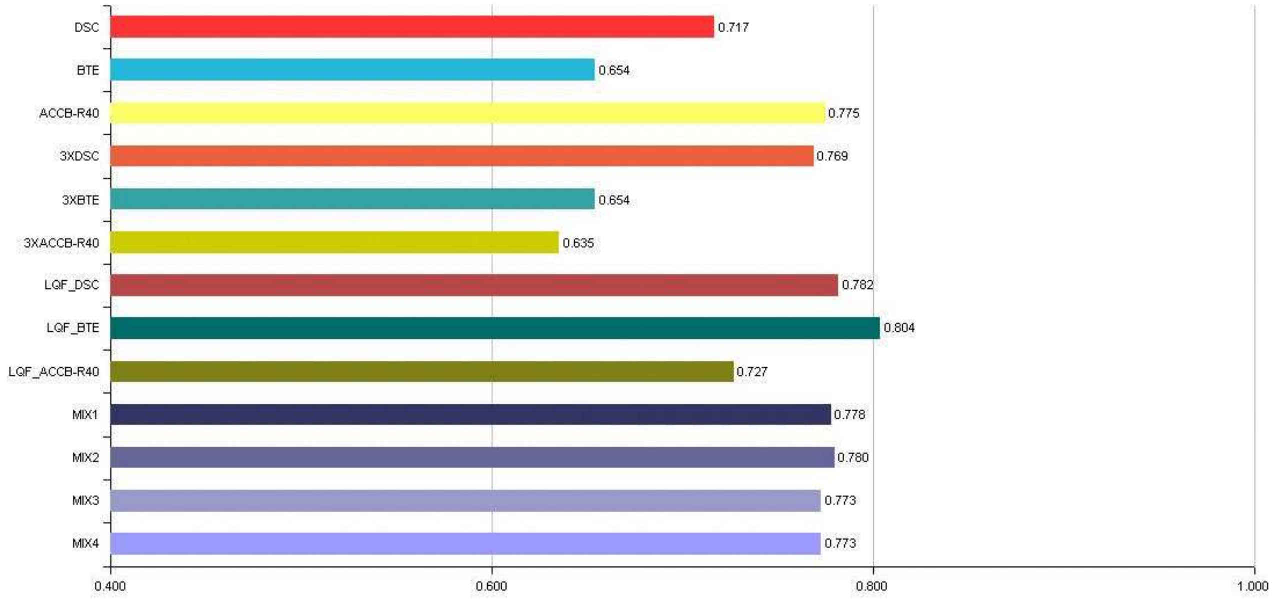


Figure 3: Diagram showing the average F1 results of the evaluation.

In a voting setup we additionally calculate the received (potentially weighted) votes for each DOM node and check whether they reach a value above the predefined threshold for inclusion in the resulting document.

5 Some first results

To evaluate the effectiveness of CE algorithm combinations, we use an existing corpus of 9.601 HTML documents. For each of those documents the main content is provided as gold standard. To compare the overlap between an extract and the gold standard we calculate the longest common (not necessarily continuous) subsequence of words in both texts. Using the number of words in the computed extract e, in the gold standard g and in their overlap $e \cap g = \mathit{lcss}(e, g)$, we can adopt the classical IR measures recall (r), precision (p) and the F1 measure (f_1) for CE evaluation:

$$r = \frac{|e \cap g|}{|g|}\ , p = \frac{|e \cap g|}{|e|}\ , f_1 = \frac{2 \cdot p \cdot r}{r + p}$$

In previous experiments we discovered the DSC, BTE, ACCB and LQF approaches to provide good results. Hence, we focused on combinations of those algorithms. Further, as LQF follows a rather different approach to CE, we constructed serial pipelines of LQF and the other three algorithms. This left us with several filter pipelines to evaluate:

3XDSC Serial execution of three DSC instances.

3XBTE Serial execution of three BTE instances.

3XACCB-R40 Serial execution of three ACCB instances.

LQF_DSC Serial setup of LQF with a downstream DSC filter.

LQF_BTE Serial setup of LQF with a downstream BTE filter.

LQF_ACCB-R40 Serial setup of LQF with a downstream ACCB filter.

MIX1 Parallel setup of DSC, BTE and ACCB with union merging of the results.

MIX2 Parallel setup of DSC, BTE and ACCB with intersection merging of the results.

MIX3 Parallel setup of DSC, BTE and ACCB with voted merging of the results, where at least two of the algorithms had to vote for an element to be included in the final document.

MIX4 Parallel setup of DSC, BTE and ACCB with weighted voted merging. DSC and BTE had a weight of 1, ACCB had a weight of 2 and the acceptance threshold was set to 2.

The detailed results of our experiments concerning the F_1 measure can be seen in table 1, a graphical representation of the average performance is shown in figure 3. Additionally to the above mentioned filter pipelines we included the performance of single BTE, DSC and ACCB filters for reference with previous results.

First of all we can observe that several combinations yield equivalent or slightly better results than a filter based on a single ACCB instance, which so far was the best general purpose CE-filter. The serial combination of LQF and BTE even delivers significantly better extracts. An explanation might be, that LQF removes navigation menus and links lists that are embedded into the main content. Hence, BTE can afterwards detect a continuous region of main content more easily. In this combination also the issue of BTE's quadratic runtime complexity is less pressing. As LQF already removes a good portion of navigation menus and alike, BTE operates on a much smaller document. Therefore, on most real-world documents (size between 60 and 100Kb) this combination takes between 1 and 1,5 seconds for extracting the main content – an acceptable performance in most application scenarios.

The parallel filters in MIX1 through MIX4 did not provide significant improvements. However, their performance is more stable across all evaluation packages and, thus, across different document styles. The triple serial execution of BTE, DSC and ACCB fostered precision at the cost of recall. Concerning F_1, this lead to improvements only for 3XDSC.

Table 1: Word Sequence F1 Results

	bbc	chip	economist	espresso	golem	heise	manual	repubblica	slashdot	spiegel	telepolis	wiki	yahoo	zdf
ACCB-R40	0.718	0.703	0.890	0.875	0.959	0.916	0.423	0.968	0.177	0.861	0.908	0.682	0.847	0.929
BTE	0.676	0.262	0.736	0.835	0.532	0.674	0.409	0.842	0.113	0.753	0.927	0.856	0.663	0.875
DSC	0.595	0.173	0.881	0.862	0.958	0.877	0.403	0.925	0.252	0.902	0.859	0.594	0.906	0.847
3XACCB-R40	0.718	0.261	0.792	0.798	0.568	0.686	0.425	0.642	0.132	0.726	0.880	0.775	0.727	0.764
3XBTE	0.676	0.262	0.736	0.835	0.532	0.674	0.409	0.842	0.113	0.752	0.927	0.854	0.663	0.875
3XDSC	0.931	0.716	0.863	0.867	0.949	0.870	0.392	0.914	0.258	0.900	0.795	0.572	0.901	0.832
LQF_ACCB-R40	0.860	0.591	0.857	0.805	0.910	0.870	0.413	0.680	0.144	0.843	0.911	0.663	0.821	0.809
LQF_BTE	0.972	0.846	0.884	0.871	0.988	0.880	0.396	0.972	0.143	0.834	0.941	0.710	0.898	0.926
LQF_DSC	0.908	0.709	0.881	0.873	0.984	0.898	0.406	0.945	0.239	0.888	0.864	0.561	0.908	0.882
MIX1	0.938	0.822	0.871	0.849	0.942	0.869	0.401	0.895	0.252	0.883	0.856	0.605	0.903	0.803
MIX2	0.936	0.824	0.872	0.866	0.954	0.868	0.380	0.915	0.252	0.893	0.854	0.596	0.891	0.823
MIX3	0.938	0.822	0.871	0.849	0.953	0.867	0.406	0.895	0.252	0.895	0.856	0.605	0.902	0.717
MIX4	0.938	0.822	0.871	0.849	0.953	0.867	0.406	0.895	0.252	0.895	0.856	0.605	0.902	0.717

6 Conclusions and Future Work

The CombinE System is capable to combine content extraction algorithms in different ways and setups. It is designed as an http-proxy which allows an easy and transparent incorporation of content filtering in IR systems accessing the web. We found some first interesting results, where filter combinations achieved better results than single filters.

A topic of ongoing and future research is the discovery and optimisation of good filter pipelines. The genetic algorithm framework for parameter optimisation [Gottron, 2009] is currently extended to explore also filter combinations. The aim is to automatically tune and further improve content extraction on HTML documents.

References

[Finn *et al.*, 2001] Aidan Finn, Nicholas Kushmerick, and Barry Smyth. Fact or fiction: Content classification for digital libraries. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001.

[Gibson *et al.*, 2005] David Gibson, Kunal Punera, and Andrew Tomkins. The volume and evolution of web page templates. In *WWW '05: Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*, pages 830–839, New York, NY, USA, 2005. ACM Press.

[Gottron, 2007] Thomas Gottron. Evaluating content extraction on HTML documents. In *ITA '07: Proceedings of the 2nd International Conference on Internet Technologies and Applications*, pages 123–132, September 2007.

[Gottron, 2008a] Thomas Gottron. Combining content extraction heuristics: the combine system. In *iiWAS '08: Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*, pages 591–595, New York, NY, USA, 2008. ACM.

[Gottron, 2008b] Thomas Gottron. Content code blurring: A new approach to content extraction. In *DEXA '08: 19th International Workshop on Database and Expert Systems Applications*, pages 29 – 33. IEEE Computer Society, September 2008.

[Gottron, 2009] Thomas Gottron. An evolutionary approach to automatically optimise web content extraction. In *IIS'09: Proceedings of the 17th International Conference Intelligent Information Systems*, pages 331–343, 2009.

[Gupta *et al.*, 2003] Suhit Gupta, Gail Kaiser, David Neistadt, and Peter Grimm. DOM-based content extraction of HTML documents. In *WWW '03: Proceedings of the 12th International Conference on World Wide Web*, pages 207–214, New York, NY, USA, 2003. ACM Press.

[Hirschberg, 1975] D. S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Commun. ACM*, 18(6):341–343, 1975.

[Mantratzis *et al.*, 2005] Constantine Mantratzis, Mehmet Orgun, and Steve Cassidy. Separating XHTML content from navigation clutter using DOM-structure block analysis. In *HYPERTEXT '05: Proceedings of the sixteenth ACM conference on Hypertext and hypermedia*, pages 145–147, New York, NY, USA, 2005. ACM Press.

[Pinto *et al.*, 2002] David Pinto, Michael Branstein, Ryan Coleman, W. Bruce Croft, Matthew King, Wei Li, and Xing Wei. QuASM: a system for question answering using semi-structured data. In *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 46–55, New York, NY, USA, 2002. ACM Press.

[Prasad and Paepcke, 2008] Jyotika Prasad and Andreas Paepcke. CoreEx: content extraction from online news articles. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1391–1392, New York, NY, USA, 2008. ACM.

[Weninger and Hsu, 2008] Tim Weninger and William H. Hsu. Text extraction from the web via text-tag-ratio. In *TIR '08: Proceedings of the 5th International Workshop on Text Information Retrieval*, pages 23 – 28. IEEE Computer Society, September 2008.

Unsupervised and domain-independent extraction of technical terms from scientific articles in digital libraries

Kathrin Eichler, Holmer Hemsen and Günter Neumann
DFKI Project Office Berlin
Alt-Moabit 91c, Berlin
{kathrin.eichler, holmer.hemsen, neumann}@dfki.de

Abstract

A central issue for making the contents of documents in a digital library accessible to the user is the identification and extraction of technical terms. We propose a method to solve this task in an unsupervised, domain-independent way: We use a nominal group chunker to extract term candidates and select the technical terms from these candidates based on string frequencies retrieved using the MSN search engine.

1 Introduction

Digital libraries (DL) for scientific articles are more and more commonly used for scientific research. Prominent examples are the Association for Computing Machinery digital library or the Association for Computational Linguistics anthology. DL may easily contain several millions of documents, especially if the DL covers various domains, such as Google Scholar. The content of these documents needs to be made accessible to the user in such a way that the user is assisted in finding the information she is looking for. Therefore, providing the user with sufficient search capabilities and efficient ways of inspecting the search results is crucial for the success of a digital library. Current DL often restrict the search to a small set of meta-labels associated with the document, such as title, author names, and keywords defined by the authors. This restricted information may not be sufficient for retrieving the documents that are most relevant to a specified query.

The extraction of technical terms (TTs) can improve searching in a DL system in two ways: First, TTs can be used for clustering the documents and help the user in finding documents related to a document of interest. Second, TTs can be provided to the user directly, in the form of a list of keywords associated with the document, and help the user in getting a general idea of what a document is about. Our input documents being scientific papers, key terms of the paper can be found in the abstract. Extracting TTs from the abstract of the document only allows us to process documents efficiently, an important issue when dealing with large amounts of data.

In this paper, we propose a method for extracting TTs in an unsupervised and domain-independent way. The paper is organized as follows. In section 2 we describe the task of technical term extraction and introduce our approach towards solving this task. After a section on related work (3), section 4 is about the generation of TT candidates based on nominal group (NG) chunking. Section 5 describes the approaches we developed to select the TTs from the list of extracted NG chunks. In section 6, we present our experimental results. We describe challenges in and first results for TT categorization (section 7) and conclude with suggestions for future work in section 8.

2 Technical term extraction

The task of extracting technical terms (TTs) from scientific documents can be viewed as a type of Generalized Name (GN) recognition, the identification of single- or multi-word domain-specific expressions [Yangarber *et al.*, 2002]. Compared to the extraction of Named Entities (NEs), such as person, location or organization names, which has been studied extensively in the literature, the extraction of GNs is more difficult for the following reasons: For many GNs, cues such as capitalization or contextual information, e.g. "Mr." for person names or "the president of" for country names, do not exist. Also, GNs can be (very long) multi-words (e.g. the term "glycosyl phosphatidyl inositol (GPI) membrane anchored protein"), which complicates the recognition of GN boundaries. An additional difficulty with domain-independent term extraction is that the GN types cannot be specified in advance because they are highly dependent on the domain. Also, we cannot make use of a supervised approach based on an annotated corpus because these corpora are only available for specific domains.

Our idea for domain-independent TT extraction is based on the assumption that, regardless of the domain we are dealing with, the majority of the TTs in a document are in nominal group (NG) positions. To verify this assumption, we manually annotated a set of 100 abstracts from the *Zeitschrift für Naturforschung*[1] (ZfN) archive. Our complete ZfN corpus consists of 4,130 abstracts from scientific papers in physics, chemistry, and biology, published by the ZfN between 1997 and 2003. Evaluating 100 manually annotated abstracts from the biology part of the ZfN corpus, we found that 94% of the annotated terms were in fact in NG positions. The remaining 6% include TTs in verb positions, but also terms occurring within an NG, where the head of the NG is not part of the TT. For example, in the NG "Codling moth females", the head of the noun group ("females") is not part of the TT ("Codling moth"). Focussing our efforts on the terms in NG position, the starting point of our method for extracting terms is an algorithm to extract nominal groups from a text. We then classify these nominal groups into TTs and non-TTs using frequency counts retrieved from the MSN search engine.

[1] http://www.znaturforsch.com/

3 Related work

3.1 NE and GN recognition

NE and GN recognition tasks have long been tackled using supervised approaches. Supervised approaches to standard NE recognition tasks (person, organization, location, etc.) have been discussed in various papers, e.g. [Borthwick *et al.*, 1998] and [Bikel *et al.*, 1999]. A supervised (SVM-based) approach to the extraction of GNs in the biomedical domain is presented by [Lee *et al.*, 2003]. Since a major drawback of supervised methods is the need for manually-tagged training data, people have, during the last decade, looked for alternative approaches. Lately, bootstrapping has become a popular technique, where seed lists are used to automatically annotate a small set of training samples, from which rules and new instances are learned iteratively. Seed-based approaches to the task of learning NEs were presented by, e.g. [Collins and Singer, 1999], [Cucerzan and Yarowsky, 1999], and [Riloff and Jones, 1999]. [Yangarber *et al.*, 2002] present a seed-based bootstrapping algorithm for learning GNs and achieve a precision of about 65% at 70% recall, evaluating it on the extraction of diseases and locations from a medical corpus. Albeit independent of annotated training data, seed-based algorithms heavily rely on the quality (and quantity) of the seeds. As lists of trusted seeds are not available for all domains, extracting GNs in a completely domain-independent way would require generating these lists automatically. A different approach, which does not rely on seeds, is applied by [Etzioni *et al.*, 2005], who use Hearst's [Hearst, 1992] list of lexico-syntactic patterns (plus some additional patterns) to extract NEs from the web. The patterns are extended with a predicate specifying a class (e.g. City) to extract instances of this particular class. The extracted instances are validated using an adapted form of Turney's [Turney, 2001] PMI-IR algorithm (point-wise mutual information). This allows for a domain-independent extraction of NEs but only from a huge corpus like the internet, where a sufficient number of instances of a particular pattern can be found. Also, using this approach, one can only extract instances of categories that have been specified in advance.

3.2 Keyword extraction

The goal of keyword extraction from a document is to extract a set of terms that best describe the content of the document. This task is closely related to our task; however, we aim at extracting *all* TTs rather than a subset. Like NE/GN recognition, keyword extraction was first approached with supervised learning methods, e.g. [Turney, 2000] and [Hulth, 2003]. [Mihalcea and Tarau, 2004] propose to build a graph of lexical units that are connected based on their co-occurrence and report an F-measure of 36.2 on a collection of manually annotated abstracts from the Inspec database. [Mihalcea and Csomai, 2007] identify important concepts in a text relying on Wikipedia as a resource and achieve an F-measure of 54.63. However, limiting the extracted concepts to those found in Wikipedia is problematic when working on specialized texts. Evaluating the annotated technical terms of the GENIA (Technical Term) Corpus, an annotated corpus of 2000 biomedical abstracts from the University of Tokyo[2], we found that only about 15% of all annotated terms (5.199 out of 34.077) matched entries in Wikipedia.

4 NG chunking

As TTs are usually in noun group positions, we extract candidates using a nominal group (NG) chunker, namely the GNR chunker developed by [Spurk, 2006]. The advantage of this chunker over others is its domain-independence, due to the fact that it is not trained on a particular corpus but relies on patterns based on closed class words (e.g. prepositions, determiners, coordinators), which are the same in all domains. Using lists of closed-class words, the NG chunker determines the left and right boundaries of a word group and defines all words in between as an NG. However, the boundaries of a TT do not always coincide with the boundaries of an NG. For example, from the NG "the amino acid", we want to extract the TT "amino acid". Therefore, we made some adaptations to the chunker in order to eliminate certain kinds of pre-modifiers. In particular, we made the chunker to strip determiners, adverbs, pronouns and numerals from the beginning of an NG. We also split coordinated phrases into their conjuncts, in particular comma-separated lists, and process the text within parentheses separately from the text outside the parentheses.

Evaluating the NG chunker for TT candidate extraction, we ran the chunker on two sets of annotated abstracts from the biology domain (ZfN and GENIA) and a set of 100 abstracts extracted from the DBLP[3] database (computer science), which was hand-annotated for TTs. To evaluate the chunker on the GENIA data, we first had to identify the annotated terms in NG position. Considering all terms with PoS tags[4] matching the regular expression $JJ^*NN^*(NN|NNS)$ as NG terms, we extracted 62.4% of all terms (57,845 of 92,722). Table 1 shows the performance of the NG chunking component of our system, evaluated on the annotated TTs in NG position of the three corpora.

	NG TTs	total matches	partial matches
ZfN	2,001	1,264 (63.2%)	560 (28.0%)
DBLP	1,316	897 (68.2%)	412 (31.3%)
GENIA	57,845	45,660 (78.9%	10,321 (11.9%)

Table 1: Evaluation of NG chunking on annotated corpora

The high number of partial matches in all corpora might be surprising; however, in many cases, these partial matches, even though untagged by the annotator, constitute acceptable TT candidates themselves. Some are due to minor variances between manual annotation and chunking, e.g. a missing dot at the end of the TT "Ficaria verna Huds." in the chunking output, or due to the fact that the extracted NG chunk is a super- or sub-NG of the annotated NG term. Common causes for partial matches are:

1. missing prepositional postmodifier, e.g. "biodegradation" and "Naphthalene" (NGs) vs. "Biodegradation of Naphthalene" (TT)
2. additional premodifiers, e.g. "new iridoid glycoside" (NG) vs. "iridoid glycoside" (TT)
3. appositive constructions, e.g. "endemic Chilean plant Latua pubiflora" (NG) vs. "Latua pubiflora" (TT)

[2] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/

[3] http://www.informatik.uni-trier.de/ ley/db/

[4] PoS tag annotation follows the Penn Treebank tagging scheme

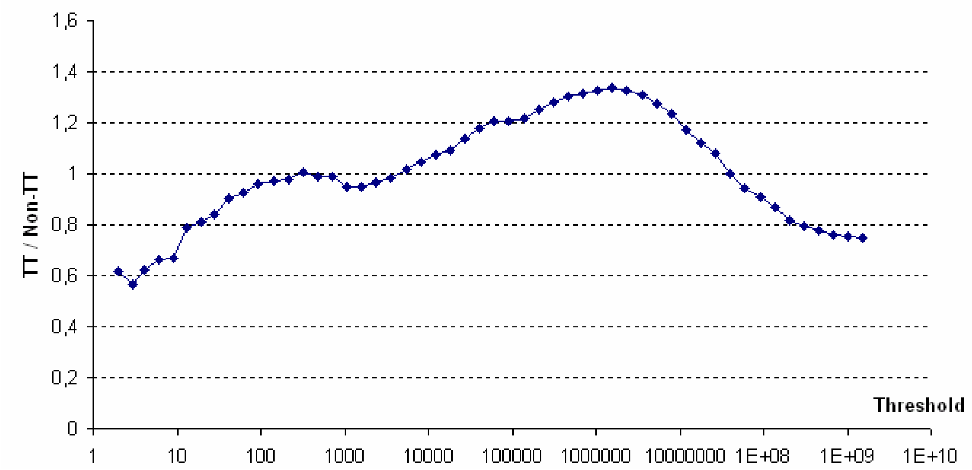


Figure 1: Ratio between TTs and non-TTs (ZfN corpus)

Real chunking errors are usually due to leading or trailing verbs, e.g. “induce hemolysis” (extracted) vs. “hemolysis” (TT). To deal with these extraction errors, we are currently evaluating methods to improve the TT candidate extraction component by learning domain-specific extraction patterns from the target corpus in an unsupervised way to supplement the domain-independent extraction patterns currently applied by the GNR.

5 Selection of technical terms

5.1 Seed-based approach

Our first approach towards determining, which of the extracted NGs are in fact TTs, was to use Wikipedia for validating part of the extracted chunks (i.e. those that constitute entries in Wikipedia, about 8% of the terms in our annotated abstracts) and use these validated chunks as seeds to train a seed-based classifier. To test this approach, we used DBpedia [Auer *et al.*, 2007] (a structured representation of the Wikipedia contents) to validate the chunks and used the validated chunks as seeds for training a seed-based GN Recognizer implemented by [Spurk, 2006]. Seed lists were generated in the following way: We first looked up all extracted NG chunks in DBpedia. For DBpedia categories, we generated a list of all instances having this category, for instances, we retrieved all categories the instance belonged to. For each category candidate, for which at least two different instances were found in our corpus, we then created a seed list for this category, containing all instances found for this category in DBpedia. For each instance candidate, we generated seed lists for each category of the instance accordingly. These lists were used as positive evidence when training the seed-based GN Recognizer. In addition, we used seed lists containing frequent words, serving as negative evidence to the learner. Our frequent word seed lists were generated from a word frequency list based on the British National Corpus[5]. From this list, we extracted each word together with its PoS tag and frequency. After preprocessing the data (i.e. removing the “*” symbol at the end of a word and removing contractions), we generated a list of words for each PoS tag separately.

An evaluation of the seed-based GN learner on the ZfN corpus (4,130 abstracts) showed that the results were not satisfying. Learning to extract instances of particular categories, the number of found sample instances in the corpus was too small for the learner to find patterns. Experiments on learning to extract instances of a general type ”technical term” showed that the TTs are too diverse to share term-inherent or contextual patterns.

In particular, the use of DBpedia for the generation of seed lists turned out unpractical for the following reasons: 1. DBpedia is not structured like an ontology, i.e. instances and categories are often not in an is-a-relation but rather in an is-related-to-relation. For example, for the category “protein”, we find instances that are proteins, such as “Globulin”, but we also find instances such as “N-terminus” that are related to the term “protein” but do not refer to a protein. However, as the seed-based learner relies on morphological and contextual similarities among instances of the same type when trying to identify new instances, better results could only be achieved using a knowledge base, in which instances and categories are structured in a clearly hierarchical way. 2. Seed-based learning only makes sense for ”open-class” categories. However, for some categories that we extracted from DBpedia, a complete (or almost complete) list of instances of this category was already available. For example, for the category “chemical element”, we find a list of all chemical elements and will hardly be able to find any new instance of this category in our input texts. In addition, we found that a number of terms that appeared as entries in DBpedia were in fact too general to be considered TTs, i.e. an entry such as “paper”.

5.2 Frequency-based approach

As the seed-based approach turned out unfeasible for solving the task at hand, we decided to identify the TTs within the extracted NG chunks using a frequency-based approach instead. The idea is to make use of a model introduced by [Luhn, 1958], who suggested that mid-frequency terms are the ones that best indicate the topic of a document, while very common and very rare terms are less likely to be topic-relevant terms. Inspired by Luhn’s findings, we make the assumption that terms that occur mid-frequently in a large corpus are the ones that are most associated with some topic and will often constitute technical terms. To test our hypothesis, we first retrieved frequency scores for all NG chunks extracted from our ZfN corpus of abstracts from the biology domain and then calculated the ratio between TTs and non-TTs for particular maximum frequency

[5] http://www.natcorp.ox.ac.uk/

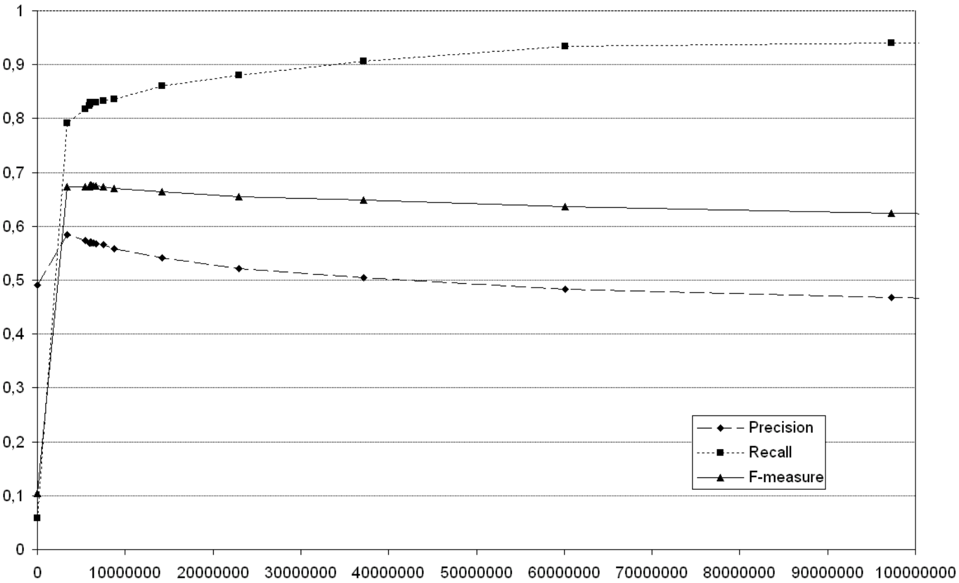


Figure 2: Optimization of t_u based on F-measure maximization (ZfN corpus)

scores. To retrieve the frequency scores for our chunks, we used the internet as reference corpus, as it is general enough to cover a broad range of domains, and retrieved the scores using the Live Search API of the MSN search engine[6]. The results, presented in Figure 1 on a logarithmic scale, confirm our hypothesis, showing that the ratio increases up to an MSN score of about 1.5 million and then slowly declines. This means that chunks with a mid-frequency score are in fact more likely to be TTs than terms with a very low or very high score.

Selecting the terms that are most likely to be TTs requires the determination of two thresholds: the lower threshold t_l and the upper threshold t_u for classifying a term candidate c with an MSN score $msn(c)$ as TT or non-TT:

$$class(c) = \begin{cases} TT & \text{if } t_l <= msn(c) <= t_u \\ nonTT & \text{elsewhere} \end{cases} \quad (1)$$

To optimize these two thresholds, we maximized the F-measure achieved on the ZfN corpus with different thresholds set. For t_l, we simply tried all thresholds from 0 to 10 and found a threshold of 1 to yield the best results. This might seem surprising; however, as many technical terms are in fact retrieved only once or twice by MSN, recall drops dramatically very fast if a higher value of t_l is chosen. For t_u, rather than trying out all numbers from 1 to several million, we used a simple but robust optimization algorithm - golden-section search [Kiefer, 1953] - to converge towards a (local) optimum threshold. Using this method, we determined an upper threshold of 6.05 million (cf. Figure 2) for the ZfN corpus. In order to find out whether this threshold is different for other domains, we applied the same method to optimize the threshold for the DBLP corpus (computer science). For this corpus, the maximum F-measure was achieved with a threshold of about 20 million. We are currently developing methods for determining this threshold automatically, without using annotated training data.

6 Experimental results

Evaluating our algorithm on our three annotated corpora of abstracts, we obtained the results summarized in Table 2. The scores for the ZfN corpus are comparable to results for GN learning, e.g. those by [Yangarber *et al.*, 2002] for extracting diseases from a medical corpus. For the DBLP corpus, they are considerably lower, which can be explained by the fact that terminology from the computer science domain is much more commonly found in the internet than terminology from other domains. This results in a greater overlap of TTs and non-TTs with similar MSN frequencies and, consequently, in lower classification performance.

To evaluate our approach in an unsupervised way (i.e. without using the annotated corpora for threshold optimization), we selected the top half[7] of the extracted NG chunks as TTs and compared this list to the set of annotated TTs and to a set of the top half of extracted NG chunks selected using TF/IDF, a baseline measure commonly applied in keyword extraction. As “top half”, we considered the chunks with the lowest MSN score (with an MSN score of at least 1) and those chunks with the highest TF/IDF score, respectively. The results, summarized in Table 3, show that our MSN-based method yields considerably better results than the TF/IDF baseline. The F-measure of 0.55 for terms in NG position corresponds to the score achieved by [Mihalcea and Csomai, 2007] for Wikipedia terms. However, our method does not limit the extracted terms to those appearing as entries in the Wikipedia encyclopedia.

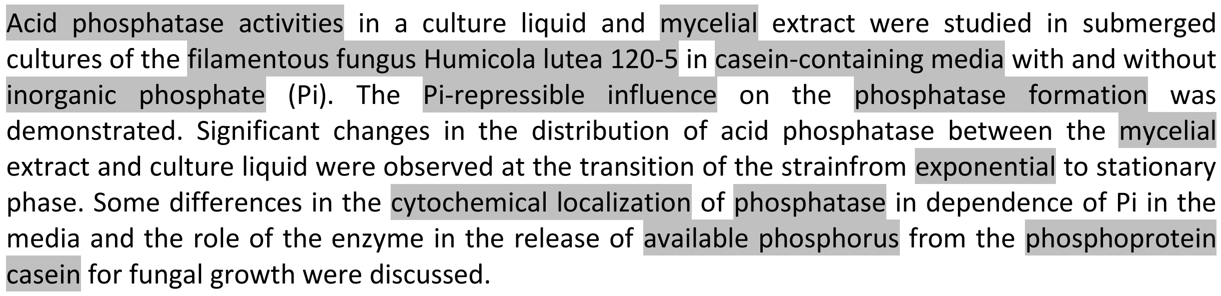
Figure 3 shows a sample abstract from the ZfN corpus, with the identified TTs shaded.

7 Categorization of technical terms

In contrast to classical NE and GN recognition, our approach does not automatically perform a categorization of the extracted terms. For a domain-independent approach towards categorization, we have analyzed the use of DBpedia. Every instance found in DBpedia has one or more

[6] http://dev.live.com/livesearch/

[7] Analysing our different corpora, we found that the number of TTs annotated in a text is usually about half the number of extracted NGs



Acid phosphatase activities in a culture liquid and mycelial extract were studied in submerged cultures of the filamentous fungus Humicola lutea 120-5 in casein-containing media with and without inorganic phosphate (Pi). The Pi-repressible influence on the phosphatase formation was demonstrated. Significant changes in the distribution of acid phosphatase between the mycelial extract and culture liquid were observed at the transition of the strainfrom exponential to stationary phase. Some differences in the cytochemical localization of phosphatase in dependence of Pi in the media and the role of the enzyme in the release of available phosphorus from the phosphoprotein casein for fungal growth were discussed.

Figure 3: ZfN sample output of the TT extraction algorithm

	Precision	Recall	F1
ZfN (biology)	58%	81%	0.68
DBLP (computer science)	48%	65%	0.55
GENIA (biology)	50%	75%	0.60
Yangarber (diseases)	65%	70%	0.67

Table 2: Evaluation of TT extraction on annotated corpora

	Precision	Recall	F1
GENIA NG terms only (vs. all GENIA terms)			
GNR + MSN	51% (56%)	61% (47%)	0.55 (0.51)
GNR + TF/IDF	45% (51%)	53% (42%)	0.49 (0.46)

Table 3: Comparison to TF/IDF baseline

categories associated. However, the problems of using DBpedia for categorization are

1. to identify the correct domain, e.g. “vitamin C” is related to categories from the biology domain, but also to categories from the music domain
2. to choose an appropriate category if several categories of the same domain are suggested, e.g. “vitamin C” belongs to categories “vitamins”, “food antioxidants”, “dietary antioxidants”, “organic acids”, etc.
3. to identify the specificity of the category, e.g. the term “Perineuronal net” is linked to the categories “Neurobiology” and “Neuroscience”, where “Neurobiology” also appears as subcategory of “Neuroscience”.
4. to categorize instances not found in DBpedia.

To deal with the first two problems, we have evaluated a PMI/IR-based approach, using Turney’s [Turney, 2001] formula to determine the best category for a given instance in a particular context. Turney computes the semantic similarity between an instance and a category in a given context by issuing queries to a search engine. The score of a particular choice (in our case: one of the categories) is determined by calculating the ratio between the hits retrieved with a problem (in our case: the instance) together with the choice and a context (in our case: other terms in the input text) and hits retrieved with the choice and the context alone. For evaluating our algorithm, we retrieved the list of DBpedia categories for 100 of our extracted terms with an entry in DBpedia and manually chose a set of no, one or several categories fitting the term in the given context. We then ran our algorithm with three different minimum PMI/IR score thresholds (0, 0.5 and 1) set and compared the output to the manually assigned categories. We then calculated precision, recall and F1 for each of these thresholds and compared the results to two different baselines. Baseline algorithm 1 always assigns the first found DBpedia category, baseline algorithm 2 never assigns any category. The results are summarized in Table 4. Baseline 2 is calculated because only about 22% of the possible categories were assigned by the human annotator. The majority of terms (53%) was not assigned any of the proposed categories. This is because many terms that appeared as entries in DBpedia were not used as technical terms in the given context but in a more general sense. For example, the term “reuse” (appearing in a computer science document), is linked to the categories “waste management” and “technical communication”, neither of which fit the given context. Due to this proportion of assigned to non-assigned categories, a PMI/IR threshold of 0 turns out to be too low because it favors assigning a category over not assigning any category. With a threshold of 0, the combined F1 score stays below the baseline score of never assigning any category. With thresholds set to 0.5 and 1, however, the combined F1 score is considerably higher than both baselines. A threshold of 0.5 yields considerably better results for terms with one or more assigned categories and a slightly better overall result than a threshold of 1. The results show that the algorithm can be used to decide whether a proposed DBpedia category fits an instance in the given context or not. In particular, with a PMI/IR score threshold set, it can achieve high precision and recall scores when deciding that a category does not fit a term in the given context.

8 Conclusion and current challenges

We have presented a robust method for domain-independent, unsupervised extraction of TTs from scientific documents with promising results. Up to now, we are not able to categorize all extracted TTs, as is usually done in GN learning, but presented first experimental results towards solving this task. The key advantage of our approach over other approaches to GN learning is that it extracts a broad range of different TTs robustly and irrespective of the existence of morphological or contextual patterns in a training corpus. It works independent of the domain, the length of the input text or the size of the corpus, in which in the input document appears. Current challenges include improving the TT candidate extraction component, in particular the recognition of TT boundaries, in order to reduce the number of partial matches. For TT selection, our goal is to determine MSN frequency thresholds automatically, without using annotated training data. Another major challenge is the categorization of all TTs.

	Thresh = 0	**Thresh = 0.5**	**Thresh = 1**	**Baseline 1**	**Baseline 2**
Category assignment					
Precision	36.56%	50.00%	48.89%	37.00%	N/A
Recall	53.13%	43.75%	34.38%	57.81%	0.00%
F1	0.43	0.47	0.4	0.45	N/A
No category assignment					
Precision	91.67%	80.43%	71.93%	N/A	53.00%
Recall	20.75%	69.81%	77.36%	0.00%	100.00%
F1	0.34	0.75	0.75	N/A	0.69
Combined results					
Precision	42.86%	63.73%	61.76%	37.00%	53.00%
Recall	38.46%	55.56%	53.85%	31.62%	45.30%
F1	0.41	0.59	0.58	0.34	0.49

Figure 4: Evaluation of DBpedia categorization using different PMI/IR thresholds

Acknowledgments

The research project DILIA (Digital Library Assistant) is co-funded by the European Regional Development Fund (EFRE) under grant number 10140159. We gratefully acknowledge this support.

References

[Auer *et al.*, 2007] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A nucleus for a web of open data. In *Proceedings of the 6th International Semantic Web Conference*, Busan, Korea, November 2007.

[Bikel *et al.*, 1999] D. M. Bikel, R. Schwartz, and R. M. Weischedel. An Algorithm that Learns What's in a Name. *Machine Learning*, 34:211–231, 1999.

[Borthwick *et al.*, 1998] A. Borthwick, J. Sterling, E. Agichstein, and R. Grishman. NYU: Description of the MENE named entity system as used in MUC-7. In *Proceedings of the 7th Message Understanding Conference*, 1998.

[Collins and Singer, 1999] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–111, Maryland, USA, 1999.

[Cucerzan and Yarowsky, 1999] S. Cucerzan and D. Yarowsky. Language independent named entity recognition combining morphological and contextual evidence. In *Proceedings of the Joint SIGDAT Conference on EMNLP/VLC*, 1999.

[Etzioni *et al.*, 2005] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D.S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165:91–134, 2005.

[Hearst, 1992] M. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*, pages 539–545, 1992.

[Hulth, 2003] A. Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[Kiefer, 1953] J. Kiefer. Sequential minimax search for a maximum. In *Proceedings of the American Mathematical Society 4*, 1953.

[Lee *et al.*, 2003] K. Lee, Y. Hwang, and H. Rim. Two-phase biomedical NE recognition based on SVMs. In *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, 2003.

[Luhn, 1958] H.-P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2:157–165, 1958.

[Mihalcea and Csomai, 2007] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *CIKM '07*, New York, NY, USA, 2007. ACM.

[Mihalcea and Tarau, 2004] R. Mihalcea and P. Tarau. TextRank: Bringing order into texts. In *Proceedings of EMNLP-04 and the 2004 Conference on Empirical Methods in Natural Language Processing*, July 2004.

[Riloff and Jones, 1999] E. Riloff and R. Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the 16th National Conference on AI*, Orlando, FL, 1999.

[Spurk, 2006] C. Spurk. Ein minimal überwachtes Verfahren zur Erkennung generischer Eigennamen in freien Texten. Diplomarbeit, Saarland University, Germany, 2006.

[Turney, 2000] P. D. Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336, May 2000.

[Turney, 2001] P. D. Turney. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the 12th European Conference on Machine Learning*, Freiburg, Germany, 2001.

[Yangarber *et al.*, 2002] R. Yangarber, L. Winston, and R. Grishman. Unsupervised learning of generalized names. In *Proceedings of the 19th International Conference on Computational Linguistics*, 2002.

Towards Unifying Selection Mechanisms for DB- and IR-Systems

Klaus Benecke
Otto-von-Guericke University Magdeburg
39016 Magdeburg, Germany
benecke@iws.cs.uni.magdeburg.de

Abstract

With the help of an algebraic specification language and the functional programming language OCAML [Chailloux *et al.*, 2000] we introduced a new understanding of XML. For example, in our abstract XML specification we distinguish in the data structure and not only in the DTD between a tuple and a collection. Further all intermediate results of the generating operations are considered as XML documents. Because of this understanding of XML, we could define and implement new, powerful, and easy to handle operations for XML documents. For example, we have a restructuring operation, by which an XML document can be transferred to another one only by giving the DTD of the desired document. The paper presents a description of a complex selection operation with simple syntax, which can be used in database and information retrieval environments. It will be evident that some well known commuting rules from other data models fail, but nevertheless a first outlook for query optimization strategies is given. It will be clear that OttoQL (our language based on the discussed operations) differs from XQuery very significantly.

1 Introduction

Our end-user computer language *OttoQL* (OsTfälisch Table Oriented) [Benecke and Schnabel, 2009] was originally designed as a database language for non-first-normal-form relations. With the introduction of XML it was generalized to (XML) documents. The highlights of *OttoQL* are:

1. It is based on an algebraic specification language, which distinguishes, for example, between tuples and collections.
2. The program logic of the kernel is very simple; the operations are applied one after the other.
3. There are complex, powerful operations on structured data, which are easy to handle.

This paper covers only a part of the possibilities of *OttoQL*, namely - the select operation. The input of an *OttoQL* program can be either an XML document with DTD or a tab file. Both objects are internally represented by an (abstract) OCAML term. Such terms are transformed into new terms by operations of the program. The resulting term of the program can be represented as tab file, table, XML document, HTML document, or as an OCAML term.

```
<<L(NAME,  FIRST,L(SUBJECT,L(MARK))):: 
    Meier  Hans    Maths     1 1 1
    Schulz Michael German    1 4 4
                             4 4 4
                   Maths     1 4
    Mayer  Fritz   Maths >>
```

Table 1: pupils.xml as tab file pupils.tab

The XML document pupils.xml in Figure 1 contains data about three pupils. It can be represented in the following way in form of a tab file (Table 1).

The computer internal representation of this tab file may also include the "invisible" above tags PUPILS, PUPIL, and SUBJECTTUP. The use of these tags in the tabular representation would damage the tabular view. We will use this representation in this paper because it makes our understanding of XML more visible. Tuples are arranged horizontally and elements of collections (except at deepest level, if this level contains only one elementary field) vertically. The second section describes the specification of schemes of tabments and the specification of tabments. The *ext* operation, by which computations can be realized, is then briefly given in section 3. In the following section 4 the selection (*mit*-part) is introduced by examples and partially by corresponding XQuery programs. Selections are done in general by several *mit*-parts step by step. In the fifth section the essential part of the selection is reduced mainly to the *ext* operation. In section 6, the failure of well known rules for query optimization is shown. Furthermore, definitions are presented, which are already implicitly contained in the definition of section 5. Then unproved commuting rules, in part of a new kind, are presented. These rules are a basis for query optimization strategies (section 7). Section 8 compares shortly our approach with others and in section 9 a summary of our data model is given.

2 XML IN OCAML

In this section we present our understanding of XML in the syntax of OCAML. An XML document is also called tabment (TABle+docuMENT).

```
type coll_sym = Set | Bag | List | Any
  | S1 ;;              (* collection types: M, B, L, A, ? *)

type name = string;;                      (*column names *)

type scheme =                     (* schemes of documents *)
  Empty_s                                 (* empty scheme *)
  | Inj of name                (* each name is a scheme *)
  | Coll_s of coll_sym*scheme
                                  (*schemes for collections*)
```

```
<PUPILS>
   <PUPIL>
        <NAME>Meier</NAME>
        <FIRST>Hans</FIRST>
        <SUBJECTTUP>
            <SUBJECT>Maths</SUBJECT>
            <MARK>1</MARK>
            <MARK>1</MARK>
            <MARK>1</MARK>
        </SUBJECTTUP>
   </PUPIL>
   <PUPIL>
        <NAME>Schulz</NAME>
        <FIRST>Michael</FIRST>
        <SUBJECTTUP>
            <SUBJECT>German</SUBJECT>
            <MARK>1</MARK>
            <MARK>4</MARK>
            <MARK>4</MARK>
            <MARK>4</MARK>
            <MARK>4</MARK>
            <MARK>4</MARK>
        </SUBJECTTUP>
        <SUBJECTTUP>
             <SUBJECT>Maths</SUBJECT>
             <MARK>1</MARK>
            <MARK>4</MARK>
        </SUBJECTTUP>
   </PUPIL>
   <PUPIL>
        <NAME>Mayer</NAME>
        <FIRST>Fritz</FIRST>
        <SUBJECTTUP>
            <SUBJECT>Maths</SUBJECT>
        </SUBJECTTUP>
    </PUPIL>
</PUPILS>
```

Figure 1: Sample file pupils.xml

```
  | Tuple_s of scheme list          (* schemes for tuples *)
  | Alternate_s of scheme list;;    (* schemes for choice *)

type value =        (* disjoint union of elementary types *)
  Bar               (* a dash; only for school of interest *)
  | Int_v of big_int          (* each big integer is a value *)
  | Float_v of float
  | Bool_v of bool
  | String_v of string;;

type tabment =         (* type for tables resp. documents *)
  Empty_t                   (* empty tabment: error value *)
  | El_tab of value  (* an elementary value is a tabment *)
  | Tuple_t of tabment list             (* tuple of tabments *)
  | Coll_t of (coll_sym * scheme) *
     (tabment list)                (* collection of tabments *)
  | Tag0 of name * tabment
                          (* a tabment is enclosed by a name *)
  | Alternate_t of (scheme list) * tabment;;
   (* the type of the tabment is changed to a choice type *)
```

Examples: The string ”Hallo” (”XML document” without root) can be represented by the OCAML term

```
El_tab(String_v "Hallo")
```

and the XML document

```
<X><A>a</A><A>b</A></X>
```

can be represented for example by

```
Tag0("X",Tuple_t[
   Tag0("A",El_tab(String_v "a"));
   Tag0("A",El_tab(String_v "b"))])
```

or by

```
Tag0("X",Coll_t((List, Inj "A"),
  [Tag0("A",El_tab(String_v "a"));
   Tag0("A",El_tab(String_v "b"))])).
```

The third pupil of the above XML file has the following description:

```
Tag0 ("PUPIL",
 Tuple_t [
  Tag0 ("NAME",
    El_tab (String_v ("Mayer")));
  Tag0 ("FIRST",
    El_tab (String_v ("Fritz")));
  Coll_t ((Set,Inj "SUBJECTTUP"),[
    Tag0 ("SUBJECTTUP",
     Tuple_t [
       Tag0 ("SUBJECT",
         El_tab (String_v ("Maths")));
       Coll_t ((Set,Inj "MARK"), [ ])
      ])])])
```

We summarize the differences between the common understanding XML documents and the specified tabments:

1. The specification does not distinguish between XML-attributes and XML-elements; an attribute is signaled by a preceding ”@”.
2. Unlike to XML a tabment need not to have a root tag.
3. In the tabment, and not only in the scheme specification, a tuple of several elements is distinguished from a collection of these elements. This is an advantage, for the specification and implementation of our powerful tabment operations (restructuring *stroke*, selection, extension *ext*, *vertical*, ...).

```
   L
   |
  (X,     L)
          |
        ( Y,      L)
                  |
                  Z
```

Figure 2: Graph of the scheme L(X,L(Y,L(Z)))

4. The specification handles beside lists (L) additional proper collection types: Set (M), Bag (B), and Any (A). The ”collection” type S1 is not a proper collection. It is used for optional values (?).

3 THE ext OPERATION IN SHORT

In this paper we need only *ext* commands of the following type:

```
ext n := e at sns
```

Here *n* is a name, *e* is an expression and *sns* is a list of slashed names. If the value *v* of *e* is *Tag0(n',v')* then the given table is extended by *Tag0(n,v')* right beside each occurrence of a name of *sns*. Otherwise, it is extended by *Tag0(n,v)*.

Program 1: The syntax for addition of two columns is independent of the given structure of the tabment:

```
<< L(X, L(Y, L(Z)))::
     1    2    3 4
          5 >>
ext W := X + Y at Y
```

Result:

```
<<L(X, L(Y, W, L(Z)))::
    1    2  3    3 4
         5  6 >>
```

Here, the system extends beside Y. In order to compute the value of $X + Y$, the term $X + Y$ is occupied (substituted) stepwise during going in depth of the given tabment. If we occupy the expression by the (first) element of the given tabment then an expression $1 + Y$ results. Only if we go deeper into the Y level, complete $X + Y$ values arise. If we occupy $1 + Y$ with $<< Y :: 2 >>$ then $1 + 2$ results, and if we occupy it by the second line, then $1 + 5$ results. Because at X-level no $X + Y$ value exists by an extension

```
ext W:=X+Y at X
```

only empty extensions would arise. If we consider the extension

```
ext W:= X+Y at Z
```

then the value $1 + 5$ cannot be carried by a Z-value. If the user would not specify the *at*-part in the above Program 1 then the system generates it. The depths of the levels of the given XML document can be seen in Figure 2.

4 THE SELECTION OPERATION BY EXAMPLES

A selection is described by a *mit* part (”mit” corresponds to ”where” from SQL) or a *ohne* part (”ohne” is German, too, and means ”without”). A *mit* part has the following form: *mit [level_selector] condition*, where *level_selector* is *[slashed_names::]* or *[slashed_names:]*. Unlike some other languages a selection can be placed at an arbitrary position in an *OttoQL* program. A selection keeps the structure and thus the whole *OttoQL* DTD (document type definition) of the given tabment unchanged. The simplest *mit* part starts with the keyword *mit*, optionally followed by a level selector *n2::*, followed by a Boolean expression. All (sub-) elements of highest collections, containing this name *n2* as a component, which do not satisfy the condition, are eliminated from the tabment. Since our data are in general of a more complex structure than flat relations, we possess finer selection mechanisms. Nevertheless we have a simple syntax. The *OttoQL* DTD of *pupils.xml* contains three collection symbols, therefore we can select in three different levels:

```
mit PUPIL:: FIRST="Michael"
```

selects pupils

```
mit SUBJECTTUP:: MARK=3
```

selects SUBJECTTUP elements (all, in which a mark 3 exists) and

```
mit MARK:: MARK>3
```

selects only MARKs.
Sometimes no outer tag like *PUPIL* or *SUBJECTTUP* is present. Therefore, we also permit column names from inside the tuple as level selectors (topmost level of the corresponding collection).

```
NAME::FIRST="Michael"
```

or

```
FIRST::FIRST="Michael"
```

expresses the same as the first condition. Both conditions select pupils likewise.

Program 2: Selection at top level (Give all pupils, who have a German entry:)

```
aus doc("pupils.xml")
mit PUPIL:: SUBJECT="German"
```

The condition selects from the collection of all pupils and not from the collections of SUBJECTTUP elements. However, with each qualified pupil all its subordinated SUBJECTUP- and MARK-values appear. Thus we get the following result:

```
<<L(NAME,  FIRST,L(SUBJECT,L(MARK)))::
     Schulz Michael German    1 4 4
                              4 4 4
                   Maths      1 4 >>
```

We recognize that the condition selects all pupils, who have at least one German entry. However, all subjects of those pupils are in the result. For people, who do not know the SUBJECT tag we plan to implement also a keyword search of type $PUPIL$: ”*German*” or simply ”*German*”. If we want only all German entries then this can be easily expressed:

Program 3: Selection at second level: Find for **each** pupil (more general: with a certain property) all German data:

```
aus doc("pupils.xml")
mit SUBJECT:: SUBJECT="German"
```

Result:

```
<<L(NAME,  FIRST,L(SUBJECT,L(MARK)))::
     Meier  Hans
     Schulz Michael German    1 4 4
                              4 4 4
     Mayer  Fritz >>
```

If we apply both conditions, one after the other, then we get all *German* entries and, in addition, only the pupils, who have a *German* entry.

Program 4: Selection at two different levels:

```
aus doc("pupils.xml")
mit NAME::SUBJECT="German"
mit SUBJECT::SUBJECT="German"
```

Result:

```
<<L(NAME,  FIRST,L(SUBJECT,L(MARK)))::
    Schulz Michael German     1 4 4
                              4 4 4>>
```

If we consider the Boolean expression

```
MARK>3
```

then one can select in three collection types. In place of the following three conditions

```
mit NAME::MARK>3 #pupils with a mark>3
mit SUBJECT::MARK>3
mit MARK::MARK>3 #marks greater than 3
```

we can write shorter

```
mit NAME,SUBJECT,MARK::MARK>3
```

or even shorter as in

Program 5: Relational selection with structured output:

```
aus doc("pupils.xml")
mit MARK>3    # or: mit MARK:MARK>3
```

Result:

```
<<L(NAME,  FIRST,L(SUBJECT,L(MARK)))::
    Schulz Michael German     4 4 4 4 4
                   Maths      4 >>
```

The condition $MARK : MARK > 3$ expresses that the Boolean expression $MARK > 3$ is applied to all collections, which include MARK (are higher than MARK). For the formulation of Program 5 in XQuery we need 3 nested FLOWR constructs.

Program 6: Give all pupils (with all data), who have a mark 1 in Maths:

```
aus doc("pupils.xml")
mit NAME:: SUBJECT="Maths" and MARK=1
```

This query is equivalent to each of the following two XQuery-programs, which require SUBJECTUP-tags:

```
<PUPILS>{
  doc("pupils.xml")//
   SUBJECTTUP[SUBJECT="Maths"
              and MARK=1]/..}
</PUPILS>
```

```
<PUPILS>{
  doc("pupils.xml")//
   PUPIL[SUBJECTUP[SUBJECT="Maths"
              and MARK=1]]}
</PUPILS>
```

But Program 6 is not equivalent to the simpler XQuery program:

```
<PUPILS>{
  doc("pupils.xml")//
    PUPIL[.//SUBJECT="Maths"
           and .//MARK=1]}
</PUPILS>
```

The difference between *OttoQL* and *XPath* is that in the last *XPath* program SUBJECT and MARK are independent of each other, but in *OttoQL* both values are considered to adhere to each other. Nevertheless, it is no problem to express the last *XPath* program by *OttoQL*:

Program 7: Two independent conditions, which refer to the same collection type: Give all pupils, who have a Maths-entry, and who have a one in any subject:

```
aus doc("pupils.xml")
mit NAME:: SUBJECT="Maths"
mit NAME:: MARK=1
```

Program 8: Disjunction of two conditions:

```
aus doc("pupils.xml")
mit NAME:: SUBJECT="Maths" or MARK=1
```

Here, also the last pupil (Mayer Fritz) appears in the result, although a corresponding MARK-value does not exist. The same holds even if both names are not on a hierarchical path.

Program 9a: A condition, which contains two names, which are not on a hierarchical path, can be meaningful:

```
<< M(X, M(Y), M(Z))::
     1    2     3
     4    5     6
                7 >>
mit X:: Y=2 or Z=6
```

Here, the input tabment is also the output tabment.

Program 9b: A similar condition applied to all three levels:

```
<< M(X, M(Y), M(Z))::
     1    2     3
     4    2     6
          5     7
          8     9
     8    9
     9    2 >>
mit  Y in M[2; 5] or Z in M[6; 7]
```

Result:

```
<< M(X, M(Y), M(Z))::
     1    2
     4    2     6
          5     7
     9    2 >>
```

We note that this understanding of *or* differs from a pure relational environment, where a structure of type *M(X,Y,Z)* is given. In this case the last element could not be in the result, because it would not appear in the given table. The first result line had to be replaced by "1 2 3" and instead of the second element 8 *(X,Y,Z)*-tuples (the "8" and "9" included) would appear in the result. If we would restructure these tuples to *X,M(Y),M(Z)*, then again the second given element would result. It is necessary to note that if we apply the condition *X::Y=2 and Z=6* on a tabment of the above type *M(X,M(Y),M(Z))* the result is always empty. Nevertheless, it is clear that we can apply both conditions, one after the other, to express the intended meaning. On the other hand we can also apply the condition

2 in $M(Y)$ and 6 in $M(Z)$

on the given tabment. *M(Y)* and *M(Z)* contrary to *Y* and *Z* are on a hierarchical path of the given scheme as the following graph of Figure 3 shows. For the next program a simple relation with 2 optional columns is given:

```
      M
      |
     (X,    M,        M)
            |         |
           (Y)       (Z)
```

Figure 3: Graph of the scheme M(X,M(Y),M(Z)))

courses.xml: L(COURSE, HOURS?, PROF?)

Program 10: Give all course tuples, which have an HOURS-entry greater 20:

```
aus doc("courses.xml")
mit COURSE::HOURS>20 # or mit HOURS>20
```

Because the corresponding XML document contains no tuple tags, the solution in *XQuery* looks relatively complicated:

```
<results>
 {for $h in doc("courses.xml")//HOURS
  where $h > 20
  return
  <tup>
   {$h/preceding-sibling::COURSE[1]}
   {$h}
   {if local-name
      ($h/following-sibling::*)="PROF"
    then $h/following-sibling::PROF[1]
    else ()}
   </tup>
 }
</results>
```

5 SELECTION MORE PRECISE

The selection is based essentially on the extension operation *ext* and the *forget* operation. In short, by *ext n e sns t* (*n:=e at sns*) besides each name from *sns*, the value of *e* tagged by *n* is inserted. By Program 8 and Program 9b it becomes clear why it is not sufficient to extend at one name only. An extension

```
ext %:=SUBJECT="Maths" or MARK=1 &&
       at SUBJECT
```

would generate only empty %-values (=undefined) if we have a subject unequal to Maths, because the *MARK*-values are invisible at *SUBJECT*-level. An extension

```
ext %:=SUBJECT="Maths" or MARK=1 &&
       at MARK
```

at the other hand could not carry a truth value, if the collection of *MARK*-values is empty. But the following *ext* operation realizes the desired extensions:

```
ext %:=SUBJECT="Maths" or MARK=1 &&
       at SUBJECT,MARK
```

In the same way the corresponding extension for Program 9b is:

```
ext %:=Y in M[2;5] or Z in M[6;7] &&
       at Y,Z
```

By such %-extensions the given table is extended at each corresponding position by one of the corresponding three truth values:

```
TRUE=Tag0("%",El_tab(Bool_v true))
FALSE= Tag0("%",El_tab(Bool_v false))
UNDEFINED=Tag0("%",Empty_t)
```

```
<< M( X, L( Y, Z))::
      1     2  3
            4  5 >>
```

Table 2: Tabment T0

The selection *sel1*, which corresponds to a *mit*-part *mit* $na2$ :: $cond2$ applied to a document $t2.xml$ can be expressed roughly by the following short hand:

```
aus  doc("t2.xml")
ext  %:= cond2  at names_ex(cond2)
   # introduction of a new %-column
select all "na2-tuples", which have
a sub- or superordinated %-value "TRUE"
forget %.
```

6 TOWARDS OPTIMIZATION RULES

Unfortunately, well known commuting rules from the relational and other data models do not hold in our data model in general. First, we have a look at Table 2 (T0), on which the failing of certain well-known commuting rules for selection can be demonstrated. In the following σ is the selection operation, which corresponds to one *mit* part.

Counter examples for commuting rules of selection:

(a) $\sigma_{X::Y=4}(\sigma_{Y::Z=3}(T0)) \neq \sigma_{Y::Z=3}(\sigma_{X::Y=4}(T0))$
The left hand side is an empty tabment, unlike the right hand side. The reason is that the condition *Y::Z=3* selects elements of the **fix level** (see below) of *X::Y=4* or in other words *Y::Z=3* **refers** (see below) to a fix level *(Y,Z)* of the quantified condition *X::Y=4*. We shall see that we can commute both conditions above in another sense. *X::Y=4* can absorb *Y::Z=3*.

(b) $\sigma_{Y::pos(Y)=1}(\sigma_{Y::Y=4}(T0))$
$\neq \sigma_{Y::Y=4}(\sigma_{Y::pos(Y)=1}(T0))$
The left hand side contains the subtuple $<< Y :: 4 >>$, $<< Z :: 5 >>$, whereas the *L(Y,Z)*-collection of the right hand side is empty. The reason is again that the condition *Y::Y=4* selects in the fix level of the position selecting condition *Y::pos(Y)=1.*

(c) $\sigma_{Y::Z=3}(\sigma_{X::L(Z)[-1]=5}(T0))$
$\neq \sigma_{X::L(Z)[-1]=5}(\sigma_{Y::Z=3}(T0))$
Here, we have again a position selecting and a content selecting condition. *L(Z)[-1]* describes the *Z*-component of the last element of the list *L(Y,Z)*. The result of the left hand side contains an inner singleton and the result of the right hand side is empty. Here, *X::L(Z)[-1]=5* refers to *(X,L(Y,Z))* and has the fix level *(Y,Z)*.

(d) $\sigma_{X::Y=2}(T0) \neq T0 \quad except \quad \sigma_{X::not(Y=2)}(T0)$
Here, *except* is the set difference. The left hand side is *T0* and the right hand side the empty set of type *M(X,L(Y,Z))*. This rule is not of importance for *OttoQL*, because our set theoretical operations like *except* are defined only for flat tabments.

To work with optimization rules we need precise definitions. We want to illustrate the level of a DTD by examples of Table 3:
level(H, dtd1) = H
level(A, dtd1) = A?, B1,M(C, D)
level(C, dtd1) = (C, D)

NAME	SCHEME=type(NAME)
TABMENT	L(A?,B1,M(C,D))
A	TEXT
B1	TEXT
C	E,F
D	M(H)
E	TEXT
F	M(G)
G	BIGINT
H	TEXT
J	TEXT

Table 3: A DTD dtd1

level(F, dtd1) = (C, D)
level(C/F, dtd1) = (C, D)
level(C//G, dtd1) = G
level(TEXT, dtd1)=(A?, B1, M(C, D))
level(M(C), dtd1) = A?, B1, M(C, D)
level(M(D)[1], dtd1) = A?, B1, M(C, D)

A condition *sname1::cond1* is called **simple** if *cond1* is a well defined Boolean valued expression and if it contains no positional attribute and each slashed name is of elementary type (TEXT, ...) and each deepest slashed name from *cond1* is at the same level as *sname1*. In this case, the condition ”contains” no (implicit) existential quantifier.

A condition is called **relational** if it is of type *sname1: cond1* and *sname1::cond1* is simple. A condition *cond* without level specification is therefore always equivalent to a relational condition if it contains only elementary slashed names. Therefore, we will call it relational, too.

In the above given tabment *T0 Y:Z=3* and *X:X=1* are relational conditions, contrary to *X:Y=4*. Again, with respect to *T0*, *Y:Z=3* is an abbreviation of *X::Z=3* followed by *Y::Z=3*. A condition can have zero, one, or several fix levels. The simple condition *Y::Z=3* (consider tabment *T0*) refers to *level(Y) = level(Z) = (Y,Z)* has no fix level. Generally, simple conditions have no fix levels. The ”quantified” condition *X::Y=4* refers to *(X,L(Y,Z))* that means it selects *(X,L(Y,Z))*-elements and has the fix level *(Y,Z)*. That means the truth value of the condition depends on the *L(Y,Z)*-collection. If *(Y,Z)*-elements are eliminated by another condition then the truth value for evaluating a *level(X)* -element may change.

A scheme *lev* is a **fix level** of a condition *sname1::cond1* if one of the following cases is satisfied:

1. *cond1* contains an attribute *C(sname)*, or *pos(sname)* with *lev=level(sname)*, or *att[i]* where *att* is of collection type and *lev=level(element-type(att))*.
2. *cond1* contains an attribute *att* such that *lev* is superordinated to *att* (or at same level) and *sname1* superordinated to *lev* and *lev* is unequal to *level(sname1)*.
3. *cond1* contains a collection name attribute *cn* and *cn* is of type *C(lev)*.

We present some illustrating examples with respect to the above DTD *dtd1*:

1. *C:: M(G)=M[1; 2]* has fix level *(G)*, but not *(C,D)* (1)
2. $C :: pos(G) < 40$ has fix level *(G)*, but not *(C,D)* (1)
3. $C :: M(G)[3] = 2$ has fix level *(G)*, but not *(C,D)* (1)
4. *A:: G=1* has fix the levels: *(G)* and *(C,D)*, (2)
   but not *(A?,B1,M(C,D))*
5. *A::1 in M(G)* has fix level *(C,D) (att=M(G))* (2)
   and *(G)*, but not *(A?,B1,M(C,D))* (1)

```
<< M( X, M(Y), M(Z))::
      1    2     2
                 3 >>
```

Table 4: Tabment T1

```
<< M( W, M( X, Y), M( X, Z))::
      1     1  1      1  1
                      2  2 >>
```

Table 5: Tabment T2

6. $D = L[1;2]$ has fix level H. (3)

The ”counter examples” presented after each of the following conjectures are not real counter examples. They are to demonstrate that we cannot omit corresponding presuppositions.

Conjecture 1 (sel-sel1):
If *sn1::c1* does not select in a fix level of *sn2::c2* and *sn2::c2* does not select in a fix level of *sn1::c1* then the following holds:

$$\sigma_{sn2::c2}(\sigma_{sn1::c1}(tab)) = \sigma_{sn1::c1}(\sigma_{sn2::c2}(tab))$$

Counter example (e):

$$\sigma_{Z::Z=3}(\sigma_{Y::Y inM(Z)}(T1)) \neq \sigma_{Y::Y inM(Z)}(\sigma_{Z::Z=3}(T1))$$

The right and left hand sides are equal to the following tabments, respectively:

```
<< M( X, M(Y), M(Z))::
      1    2     3 >>
```

```
<< M( X, M(Y), M(Z))::
      1          3 >>
```

Here, *Z::Z=3* selects in a fix level *(Z)* of *Y::Y in M(Z)*.

Conjecture 2 (sel–sel2):
If *sn1:c1* and *sn2:c2* are relational and all occurrences of attributes from *sn1*, *sn2*, *c1*, and *c2* are on one hierarchical path then the following holds:

$$\sigma_{sn2:c2}(\sigma_{sn1:c1}(tab)) = \sigma_{sn1:c1}(\sigma_{sn2:c2}(tab))$$

Constructed counter example (f):

$$\sigma_{Y:Y=1}(\sigma_{X:X=2}(T2)) \neq \sigma_{X:X=2}(\sigma_{Y:Y=1}(T2))$$

The left hand side is the empty table and the right hand side is:

```
<< M( W, M( X, Y), M( X, Z))::
      1               2  2 >>
```

Non hierarchical path counter example of ordinary type (g):

$$\sigma_{Z:Z=4 or Y=4}(\sigma_{Y:Y=2}(T3)) \\ \neq \sigma_{Y:Y=2}(\sigma_{Z:Z=4 or Y=4}(T3))$$

Here the left hand side is again empty and the right hand side is equal to:

```
<< M( X, M(Y), M(Z))::
      1    2   >>
```

Here, *Z* and *Y* are not on a hierarchical path and *Z:Z=4* or *Y=4* is not relational. But, we remark that the following equation holds, because both sides are empty.

$$\sigma_{Z=4 or Y=4}(\sigma_{Y=2}(T3)) = \sigma_{Y=2}(\sigma_{Z=4 or Y=4}(T3))$$

```
<< M( X, M( Y), M( Z))::
      1     2      3
            4 >>
```

Table 6: Tabment T3

Conjecture 3 (sel–intersect):
If *t* is a set or bag and *sn1::c1* and *sn2::c2* refer to the outmost level and *sn1::c1* is not position selecting then the following holds:

$$\sigma_{sn2::c2}(\sigma_{sn1::c1}(t)) = \sigma_{sn1::c1}(t) intersect \sigma_{sn2::c2}(t))$$

Conjecture 4 (sel–conjunction 1):
If the condition *sn::c1* does not select in a fix level of *sn::c2* and *sn::c2* does not select in a fix level of *sn::c1* then the following holds:

$$\sigma_{sn::c1andc2}(tab) = \sigma_{sn::c1}(\sigma_{sn::c2}(tab))$$

Counter example (h1):

$$\sigma_{Y::Y=2}(\sigma_{Y::4inL(Y)}(T0))$$
$$\neq \sigma_{Y::4inL(Y)}(\sigma_{Y::Y=2}(T0))$$
$$\neq \sigma_{Y::4inL(Y)andY=2}(T0))$$

The first and third expression is equal to the first following tabment and the second to the second following:

```
<< M( X, L( Y, Z))::
      1     2  3>>

<< M( X, L( Y, Z))::
      1 >>
```

Counter example (h2):

$$\sigma_{X::Y=2}(\sigma_{X::Z=5}(T0))$$
$$\neq \sigma_{X::Z=5}(\sigma_{X::Y=2}(T0))$$
$$\neq \sigma_{X::Y=2andZ=5}(T0)$$

The result of the first two expressions is *T0* and the result of the third is empty.

Conjecture 5 (sel–conjunction 2):
If *sn:c1* and *sn:c2* are relational and all occurrences of attributes from *sn*, *c1*, and *c2* are on one hierarchical path then the following holds:

$$\sigma_{sn:c1andc2}(tab) = \sigma_{sn:c1}(\sigma_{sn::c2}(tab))$$

Conjecture 6 (sel–conjunction 3):
If *sn::c1* is not position selecting and *sn* determines the outmost level of a given set, bag, or list *tab* then the following holds:

$$\sigma_{sn::c1andc2}(tab) \text{ in2 } \sigma_{sn::c1}(\sigma_{sn::c2}(tab))$$

Here, *in2* is the set (bag) (list) theoretic inclusion.

Counter example (inequality) (i):

$$\sigma_{X::Y=2andZ=5}(T0) \neq \sigma_{X::Y=2}(\sigma_{X::Z=5}(T0))$$

Here, the left hand side is the empty tabment and the right hand side results in T0.

Conjecture 7 (absorb–sel):
If *sn1::c1* is a simple condition and *c2* contains only slashed names as attributes and *sn1::c1* refers to a fix level of *sn2::c2* then the following holds:

$$\sigma_{sn2::c2}(\sigma_{sn1::c1}(tab)) = \sigma_{sn1::c1}(\sigma_{sn2::c1andc2}(tab))$$

Counter example (absorb–sel) (j):

$$\sigma_{X::4inM(Y)}(\sigma_{Y::Y=2}(T0))$$
$$\neq \sigma_{Y::Y=2}(\sigma_{X::4inM(Y)andY=2}(T0))$$

Here, the left hand side is empty and the right hand side equal to the following tabment:

```
<< M( X, M( Y, Z))::
      1     2  3 >>
```

Conjecture 8 (::-condition-to :-condition):
Assume *sn1:c1* and *sn2:c2* are relational conditions and *sn2* is deeper than *sn1* then the following holds:

$$\sigma_{sn2:c2}(\sigma_{sn1::c1}(tab)) = \sigma_{sn2:c2}(\sigma_{sn1:c1}(tab))$$

7 A VERY SMALL OPTIMIZATION EXAMPLE

One of the main principles of database optimization strategies is to apply high selective conditions before less selective ones. Further, conditions, for which an index exist, have to be realized firstly. We consider an application of these rules for query optimization. Let *T02* be a file of type *M(X,L(Y,Z))* for which TID's exist. A promising (H2O-) file concept is described in [Benecke, 2008]. Here, records are small XML-files, which are addressed by TIDs. We consider the query:

```
aus doc("T02.h2o")
mit Y:: Z=3
mit X:: Y=4
```

Here, the first condition $Y :: Z = 3$ cannot be supported by a simple index, because the condition selects only subtuples and not tuples. That means we have to access to all records, if we want to realize this condition first. But, by *absorb–sel* the query can be transformed to:

```
aus doc("T02.h2o")
mit X:: Y=4 and Z=3
mit Y:: Z=3
```

Now, this program can be optimized to the following one, where the first two conditions could be supported by simple indexes and the corresponding address sets can be intersected because of *sel–intersect*. A simple index for *Y* contains for each *Y*-value only the TID's of the records, which contain this *Y*-value:

```
aus doc("T02.h2o")
mit X:: Y=4
mit X:: Z=3
mit X:: Y=4 and Z=3
mit Y:: Z = 3
```

8 Related work

XQuery [Boag *et al.*, 2007] is a very powerful, well understood computer language for XML files and collections of XML files. But, XQuery seems to be more complicated than SQL. Therefore, we do not believe that in future the number of XQuery users will exceed the number of SQL users. We believe that *OttoQL* is more easy to use for a broad class of queries than XQuery and even SQL. We trace this back mainly to our simple syntax. Our semantic is more complicated than the semantic of SQL.
In XAL [Frasincar *et al.*, 2002] and most other languages and algebras the select operation is in general commutative. [Li *et al.*, 2004] is an interesting article with similar aims as ours. Here, it is tried to generalize XQuery in a way that the user has not to know the structure (DTD) of the given documents in detail. Or in other words that one query can be applied to several XML documents of a similar structure. They use three queries and two DTD's to illustrate their theory. Here, the formulation of the second of three queries of [Li *et al.*, 2004] in *OttoQL* follows:

```
A: BIBLIOGRAPHY = BIB*
   BIB = (YEAR, BOOK*, ARTICLE*)
   BOOK = TITLE, AUTHOR*
   ARTICLE = TITLE, AUTHOR*
B: BIBLIOGRAPHY = BIB*
   BIB = (BOOK* | ARTICLE*)
   BOOK = YEAR, TITLE, AUTHOR*
   ARTICLE = YEAR, TITLE, AUTHOR*
```

Query 2: Find additional authors of the publications, of which Mary is an author.

```
aus  doc("bib.xml")
mit  TITLE::AUTHOR="Mary"
ohne AUTHOR::AUTHOR="Mary"
gib  M(AUTHOR)
```

In [Li *et al.*, 2004] a MCLAS (Meaningful Lowest Common Ancestor Structure) function is used to express these queries. As in our examples, the formulation of these queries does not require knowledge about the exact structure of the document and the tags BIBLIOGRAPHY, BIB, BOOK, and ARTICLE are not needed, too. But in [Li *et al.*, 2004], contrary to our approach, these tags are needed to find the ancestors.

In [Bast and Weber, 2007] IR goes one step into DB-integration. Here, Bast and Weber start with an efficient algorithm and data structure and then they develop a user interface. In *OttoQL* we started to develop a user language, example by example, and now we try to develop a theory and an efficient implementation. The CompleteSearch Engine of [Bast and Weber, 2007] is a full text retrieval system, which uses tags and the join. The system does not support aggregations and does not allow restructuring. We present the query: Which German chancellors had an audience with the pope?

```
german chancellor politician:
audience pope politician:
```

```
mit "german" and "chancellor"
gib M(politician)
intersect &&
{ mit "audience" and "pope"
  gib M(politician)}
```

9 SUMMARY OF OttoQL

We summarize the interesting features of our model:

1. Our understanding of XML is based on independent, abstract, generating operations for tuples, collections, choice, elementary values, and tags. Therefore, we could define and widely implement powerful and easy to use operations. The use of these generating operations seems to be the reason that our OCAML programs for selection, *stroke*, ... are relatively short. So, we think that short OCAML programs are good specifications of our operations.
2. The operation *stroke* (gib part) allows a restructuring of arbitrary XML documents, where only the DTD of the desired XML document is given. It is a procedural and axiomatic generalization of the restructuring operation of [Abiteboul and Bidot, 1986]. Additional to the *restruct* operation, *stroke* allows to realize aggregations on arbitrary levels, simultaneously. A non-first-normal-form predecessor version of *stroke* is described in [Benecke, 1991]. One of the best examples of *OttoQL* is query 1.1.9.4 Q4 of [D. Chamberlain et. al. (ed.), 2007]. The presented XQuery query needs more than 20 lines. With the help of *OttoQL* it can be realized in the following way:

   ```
   aus doc("bib.xml")
   gib rs rs=M(r) r=author,L(title)
   ```

3. The select operation is very powerful, but nevertheless syntactically easy to use. It is based on the *ext* operation. Therefore it can be widely applied also independently from the given structure (DTD). This will be an advantage, if we apply a select operation on a collection of XML documents of different types. Especially tuple tags and collections tags are not necessary to formulate corresponding selections.
4. Because of new optimization rules, as yet unproven, new optimization strategies have to be developed.

Acknowledgments

I would like to thank Dmitri Schamschurko and Milan Gnjatovic for the Caml-light implementation of our forerunner nested relational algebra version of *OttoQL*. Further thanks are directed to Martin Schnabel and Xuefeng Li, who improved our ongoing implementation in several important points and to Rolf Aretz, who implements now our H2O-file concept [Benecke, 2008], which is intended to replace very large XML files. Finally, I thank the anonymous referees for their valuable remarks.

References

[Abiteboul and Bidot, 1986] S. Abiteboul and N. Bidot. Non-first-normal-form relations: An algebra allowing data restructuring. *J. Comput. System Sci*, (5):361–393, 1986.

[Bast and Weber, 2007] Holger Bast and Ingmar Weber. The complete search engine: Interactive, efficient, and towards ir & db integration. pages 88–95. CIDR2007, 2007.

[Benecke and Schnabel, 2009] Klaus Benecke and Martin Schnabel. Internet server for ottoql: `http://otto.cs.uni-magdeburg.de/otto/web/`. 2009.

[Benecke, 1991] Klaus Benecke. A powerful tool for object-oriented manipulation. In *On Object Oriented Database: Analysis, Design & Construction*, pages 95–121. IFIP TC2/WG 2.6 Working Conference, July 1991.

[Benecke, 2008] Klaus Benecke. The h2o storage structure – the marriage of the tid - concept and the xml file structure: `http://www2.cs.uni-magdeburg.de/fin_media/downloads/Forschung/preprints/2008/TechReport16.pdf`. In *Reports of the faculty of computer science*, pages 1–18. University Magdeburg, December 2008.

[Boag *et al.*, 2007] S. Boag, D. Chamberlain, and D. Florescu et. al. Xquery 1.0: An xml query language: `http://www.w3.org/TR/xquery/`. *J. Comput. System Sci*, 2007.

[Chailloux *et al.*, 2000] Emmanuel Chailloux, Pascal Manoury, and Bruno Pagano. *Developing Applications With Objective Caml: `http://caml.inria.fr/oreilly-book/`.* Paris, France, 2000.

[D. Chamberlain et. al. (ed.), 2007] D. Chamberlain et. al. (ed.). Xml query use cases: `http://www.w3.org/TR/xmlquery-use-cases`. 2007.

[Frasincar *et al.*, 2002] F. Frasincar, G.-J. Houben, and C. Pau. Xal : an algebra for xml query optimization. In *ADC*, 2002.

[Li *et al.*, 2004] Y. Li, C. Yu, and H.V.Jagadish. Schema-free xquery. In *VLDB Conference*, pages 72–83, 2004.

Searching Wikipedia Based on a Distributed Hash Table
Suchen in Wikipedia basierend auf einer Distributed Hash Table

Judith Winter
Johann Wolfgang Goethe Universität
Frankfurt am Main, Deutschland
winter@tm.informatik.uni-frankfurt.de

Joscha Stützel
Fachhochschule Hamburg
Hamburg, Deutschland
joscha.stuetzel@web.de

Abstract

In diesem Papier wird ein System präsentiert, mit dem die Wikipediakollektion mittels XML Information Retrieval Techniken durchsucht werden kann, um relevante strukturierte Dokumente zu finden. Das System basiert dabei auf einer Distributed Hash Table (DHT), die über eine Menge teilnehmender Rechner verteilt ist. Die einzelnen Rechner bilden somit ein strukturiertes Peer-to-Peer (P2P) Netz, über das die zu durchsuchende Wikipediakollektion im XML-Format und geeignete Indizes verteilt sind. Die Architektur des Systems als Verbund gleichberechtigter Teilnehmer (Peers), die durch Kombination ihrer Ressourcen (Speicher, Rechenpotential) ein leistungsstarkes System bilden, entspricht dabei der Idee hinter Wikipedia: die Ressourcen (das Wissen) der einzelnen Teilnehmer (Wikipedia-Autoren) zusammenzufügen, um der Gemeinschaft eine umfangreiche Digitale Bibliothek zur Verfügung zu stellen. Die Evaluierung des präsentierten Systems zeigt, wie XML Information Retrieval Techniken dabei helfen können, geeignete Informationen aus der zugrundeliegenden DHT auszuwählen, um strukturierte Anfragen auf die Wikipediakollektion im Netz weiterzuleiten und eine Verbesserung der Suchqualität zu erzielen.

1. Einleitung

INEX, die INiative for the Evaluation of XML-Retrieval, bietet eine Plattform zum Vergleich von Systemen, die Techniken des Information Retrievals (IR) auf XML-Dokumente anwenden [Trotman *et al.*, 2009]. Es wird also in solchen Kollektionen gesucht, die mit der eXtensible Markup Language (XML) strukturiert sind [Bray *et al.*, 2006]. Die Suchqualität in Bezug auf Präzision und Recall kann dabei gesteigert werden, indem man sich die Struktur der XML-Dokumente zunutze macht. Methoden, die dabei zum Einsatz kommen, sind beispielsweise die unterschiedliche Gewichtung verschiedener XML-Elemente, das Verwenden von Element- statt Dokumentstatistiken, das Retrieval von Passagen (z.B. verschachtelte XML-Elemente) statt ganzen Dokumenten oder die Einbeziehung von Strukturhinweisen des Benutzers durch Verwenden von content-and-structure (CAS)-Anfragen [Luk *et al.*, 2002; Amer-Yahia and Lalmas, 2006]. Die Testkollektion, die aktuell bei INEX zur Evaluierung solcher Systeme verwendet wird, ist die Wikipediakollektion, wobei die einzelnen Artikel im XML-Format vorliegen [Denoyer and Gallinari, 2006].

Herkömmliche XML-Retrieval Lösungen sind teilweise bereits sehr erfolgreich bei der Suche in dieser Kollektion. Es handelt sich jedoch durchgehend um zentralisierte Systeme, die auf einzelnen Rechnern laufen. Bisher hat noch keiner der evaluierten Ansätze Verteilungsaspekte berücksichtigt, z.B. um eine technisch sehr viel leistungsstärkere Suchmaschine durch Kombination der Ressourcen und Rechenleistung einer Menge von Rechnern zu nutzen [Winter *et al.*, 2009].

Das in diesem Papier vorgestellte System ist zum jetzigen Zeitpunkt die erste XML-Suchmaschine für *verteiltes* Retrieval. Sie basiert auf einem strukturierten Peer-to-Peer (P2P) Netz, so dass sich ihr Potential aus der Summe einer Vielzahl teilnehmender, das System bildender Rechner ergibt.

P2P-Systeme sind vielversprechende selbstorganisierende Infrastrukturen, die seit einigen Jahren in zunehmendem Maße als Alternative zu klassischen Client-Server-Architekturen eingesetzt werden. Sie bestehen aus einer Menge autonomer Peers, die gleichberechtigt zusammenarbeiten können. Ihr Potential liegt in der Fähigkeit, selbstorganisierend ohne zentrale und somit ausfallgefährdete Kontrollinstanz auszukommen. Sie können daher einerseits die Grundlage eines robusten und fehlertoleranten Systems bilden, das zu einer theoretisch unendlich großen Menge teilnehmender Rechner skalieren kann, so dass bei geschickter Verteilung von Daten und auszuführenden Prozessen eine unbegrenzt große Anzahl von Ressourcen genutzt werden kann. Eine klassische P2P-Applikation ist der gemeinsame Dateiaustausch (*Filesharing*). Durch den Zusammenschluss einer großen Anzahl von Peers zu einem Gesamtsystem entstehen Bibliotheken digitaler Dokumente in einem Umfang, wie sie singuläre Systeme nur schwer leisten können [Steinmetz and Wehrle, 2005].

Zurzeit existiert Wikipedia im Internet (http://www.wikipedia.org) als Client-/Server-basierte Anwendung. Die Idee hinter Wikipedia besteht jedoch gerade in dem, was durch P2P-Systeme realisiert wird: Kombinieren einzelner Potentiale und Ressourcen (in diesem Fall das Wissen der einzelnen Wikipedia-Autoren), um der Gemeinschaft ein leistungsstarkes System zur Verfügung zu stellen.

Das vorliegende Papier präsentiert daher ein Suchsystem für Wikipedia, das auf einem P2P-Netz basiert. Die Indizes zum Speichern von Postinglisten und Dokumentstatistiken sind dabei über eine Distributed Hash Table (DHT) [El-Ansary and Haridi, 2005] verteilt. Dies ermöglicht effizienten Zugriff auf Objekte (Dokumente, Postinglisten etc.) in *log(n)* Schritten, mit *n* = Anzahl partizipierender Peers [Risson and Moors, 2004]. Das vorliegende Papier beschreibt somit ein System, das die Wikipediakollektion mittels XML Information Retrieval (XML-Retrieval) Techniken durchsuchen kann, wobei die Anfragebeantwortung mit Hilfe von über ein ganzes P2P-Netz verteilten Informationen bewerkstelligt wird.

2. Eine DHT-basierte Suchmaschine

Das propagierte System verwendet drei Indizes, die über eine DHT verteilt sind. Der Dokumentindex enthält die Dokumentstatistiken wie Termfrequenzen. Der Elementindex speichert Statistiken über Elemente, die somit analog zu Dokumenten als Ergebnis dienen können. Und der invertierte Index beinhaltet die Postinglisten aller Indexterme. Dabei wird je Term auch seine XML-Struktur gespeichert, so dass zu jedem Term eine ganze Reihe von Postinglisten verfügbar ist, nämlich je unterschiedlicher XML-Struktur eine eigene Liste. Dies ermöglicht beim Retrieval schnellen und effizienten Zugriff auf genau diejenigen Postings, die zu einer gewünschten Struktur passen. Die Kombination aus Indexterm und seiner XML-Struktur sei als *XTerm* bezeichnet.

Alle drei Indizes sind über die Menge der am System teilnehmenden Peers verteilt, wobei der jeweils für eine Informationseinheit zuständige Peer durch Anwendung einer Hashabbildung auf den Schlüssel der Informationseinheit bestimmt wird. Für zu verteilende Postinglisten ist der entsprechende Schlüssel beispielsweise der Indexterm der Postingliste sowie dessen XML-Struktur.

Lokalisiert werden die Daten dann über die DHT, die in Form eines auf Chord [Stoica *et al.*, 2003] basierenden P2P-Protokolls implementiert wurde, wobei das Protokoll an XML-Retrieval angepasst wurde. Diese Anpassung betrifft beispielsweise die Auswahl der Schlüssel, auf die die Hashabbildung angewandt wird: statt zufälliger Verteilung der Daten über das Netz werden diejenigen Informationen zusammen auf dem gleichen Peer abgelegt, die in einer Anfrage mit einer gewissen Wahrscheinlichkeit zusammen benötigt werden.

Die Beantwortung einer Anfrage besteht dann aus den nachfolgend beschriebenen Routingschritten und dem Ranking selbst.

Beim Routing, also der Weiterleitung der Anfrage zu passenden Peers, werden drei Schritte ausgeführt. Die dabei anfallenden Nachrichten seien als Routing-Request-Nachrichten bezeichnet.

Zunächst muss die Anfrage an diejenigen Peers weitergeleitet werden, die für die Postingliste der einzelnen Anfrageterme zuständig sind. Dazu müssen die passenden Postinglisten im Netz lokalisiert werden. Mit DHT-basierten Methoden kann dies effizient in *log(n)* Schritten durchgeführt werden, indem eine Hashabbildung auf den jeweiligen Anfrageterm angewandt und der für den somit errechneten Hashwert zuständige Peer ermittelt wird (*lookup*-Funktion der DHT).

Im zweiten Schritt des Routings werden aus den lokalisierten Postinglisten eines Anfrageterms passende Postings ausgewählt und mit den ausgewählten Postings der übrigen Anfrageterme abgeglichen, indem Teile der Postinglisten (z.B. die jeweils am höchsten gewichteten 500 Postings) von Peer zu Peer geschickt werden. Es können hierbei aus Effizienzgründen nicht alle Postings ausgewählt werden, da das System ansonsten bei ansteigender Anzahl von Dokumenten und entsprechendem Anstieg der Postinglistenlängen nicht skaliert – der Aufwand für das Übertragen kompletter Postinglisten zwischen den Peers zum Abgleichen der Postings wäre zu hoch. Es ist daher essentiell, dass geeignete Postings ausgewählt werden. Hier können XML-Retrieval Techniken helfen, wie später erläutert.

Für die endgültig ausgewählten Postings müssen in einem dritten Schritt diejenigen Peers identifiziert und lokalisiert werden, die die Statistiken der Dokumente und der Elemente speichern, die von den Postings referenziert werden. Diese Statistiken sind nicht direkt in den Postinglisten abgelegt, da zu jedem Dokument eine Vielzahl von Elementen gehört, deren Statistiken nicht redundant in jeder Postingliste eines Dokuments gespeichert werden sollen. Die Anfrage wird daher zu allen Peers weitergeleitet, die die entsprechenden Statistiken speichern. Im worst case, wenn alle Statistiken auf verschiedenen Peers abgelegt sind, fällt je ausgewähltem Posting eine Weiterleitung an, die als RankingRequest-Nachricht bezeichnet sei.

Beim Ranking der Dokumente und der Elemente werden wiederum XML-Retrieval Techniken verwendet, die in diesem Papier jedoch nicht weiter Thema sind.

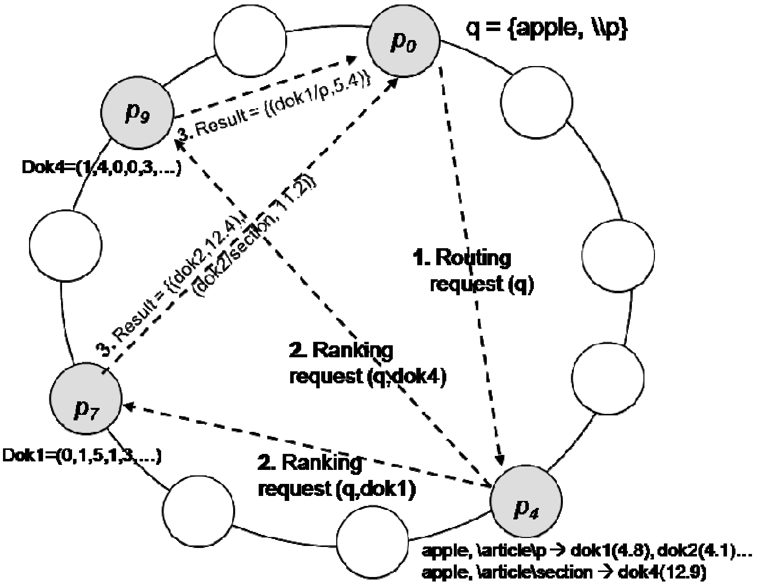

q = {apple, \\p}
p0
p9
p7
p4
3. Result = {(dok1/p,5.4)}
Dok4=(1,4,0,0,3,...)
3. Result = {(dok2,12.4)},(dok2/section,11.2)}
1. Routing request (q)
2. Ranking request (q,dok4)
Dok1=(0,1,5,1,3,...)
2. Ranking request (q,dok1)
apple, \article\p → dok1(4.8), dok2(4.1)...
apple, \article\section → dok4(12.9)

Abbildung 1: Anfragebeantwortung der Anfrage q = {apple,\\p} basierend auf einer DHT

Abbildung 1 stellt die Anfragebeantwortung einer einfachen Single-Term-Anfrage *q=(apple, \\p)* dar, wobei *q* ein einzelner XTerm ist. Dabei werde *q* durch den anfragenden Peer p_0 in einem System gestellt, das aus 11 Peers besteht. Diese seien auf einem Chord-Ring angeordnet, so dass effizientes Lookup von Informationen in *log(n)* möglich ist [Stoica *et al.*, 2003].

Gesucht wird dabei nach Absätzen (\\p steht für Paragraph) in Artikeln, deren Inhalt relevant zu dem Suchterm *apple* ist. Die Strukturangabe \\p wird dabei, analog zur inhaltsbasierten Suche mit IR-Methoden, als vager Hinweis des Benutzers gesehen, welche Strukturen von besonderem Interesse sind. Als relevante Ergebnisse werden daher auch solche angesehen, deren Struktur nicht exakt mit der angegebenen übereinstimmt.

Im Retrievalprozess wird *q* zunächst zu Peer p_4 weitergeleitet, der dem Hashwert von *apple* zugeordnet ist und somit alle Postinglisten von XTermen speichert, die den Indexterm *apple* enthalten. Peer p_4 wählt daher bei Eintreffen von *q* passende invertierte Listen aus, diese seien die für die XTerme (*apple, \article\p*) und (*apple, \article\section*), da eine starke Ähnlichkeit zwischen Absätzen (*p*) und Sektionen (*section*) angenommen wird. Aus den selektierten Postinglisten werden geeignete Postings ausgewählt, indem die Gewichte der Postings mit der Ähnlichkeit zwischen dem Struktur des XTerms und dem Strukturhinweis der CAS-Anfrage *q*, nämlich \\p, multipliziert werden. Dies begünstigt alle Postings aus Listen von XTermen, deren Struktur ähnlich zu \\p ist. Im vorliegenden Fall werden die Postings mit Dokumentreferenzen auf *dok1* und auf *dok4* ausgewählt. Daraufhin wird die Anfrage weitergeleitet zu Peer p_7 und p_9, die die entsprechenden Dokument- und Elementstatistiken der ausgewählten Postings gespeichert haben. Beide Peers erhalten die Anfrage *q*, berechnen relevante Ergebnisse und senden diese zurück an den anfragenden Peer p_0. Als Ergebnis wurden dabei auch Elemente aus den beiden ausgewählten Dokumenten errechnet.

Somit umfasst die Liste der Ergebnisse sowohl das Dokument2 (*dok2*) als auch eine Sektion aus diesem Dokument (*dok2/section*) sowie einen Absatz aus Dokument1 (*dok1/p*).

Untersucht wird im nächsten Abschnitt der Einsatz verschiedener XML-Retrieval Techniken beim Routing von strukturierten Anfragen bei der Suche in der Wikipediakollektion.

Folgende Methoden kommen dabei beim Zugriff auf die DHT und der Auswahl der Postings aus den Postinglisten der einzelnen Peers zum Tragen:

- Alle Postings werden mit einer *BM25E*-Variante [Robertson *et al.*, 2004] gewichtet, die an verteiltes XML-Retrieval angepasst wurde. Somit können unterschiedliche Elemente verschieden gewichtet werden.
- Bei der Auswahl der Postings wird die Struktur der Indexterme mit den Strukturhinweisen verglichen, die der Benutzer beim Stellen der Anfrage gegeben hat. Dies ist durch das Verwenden von CAS-Anfragen möglich. Indexterme, deren Struktur eine Mindestähnlichkeit zur vom Benutzer gewünschten Struktur aufweist, werden mit einer Erhöhung ihres Gewichts belohnt.
- Zur Bewertung fließen nicht nur Statistiken ein, die aus den durch Postings repräsentierten Dokumenten stammen, sondern auch Statistiken über die Elemente der jeweiligen Dokumente.

3. Evaluation

Das vorgestellte System wurde im Rahmen von INEX evaluiert, dazu wurde die INEX-Testkollektion des *ad-hoc* Tracks verwendet, die 4.5 Gigabytes große Wikipediakollektion, bestehend aus mehr als 650.000 XML-Dokumenten [Denoyer and Gallinari, 2006]. Abbildung 2 stellt die Suchqualität des Systems dar. Es sind dabei die gemessenen Werte für das offizielle INEX-Maß, die interpolierte Präzision im Recall-Bereich von 1% (iP[0.01]), angegeben. Verwendet wurden alle 80 Anfragen von INEX 2008, benutzt wurde der *focused* Task.

Für die Evaluierung wurde ein P2P-Netz beliebiger Größe simuliert, wobei der worst case modelliert wurde. Dazu wird angenommen, dass sich jede Postingliste eines Anfrageterms auf einem unterschiedlichen Peer befindet.

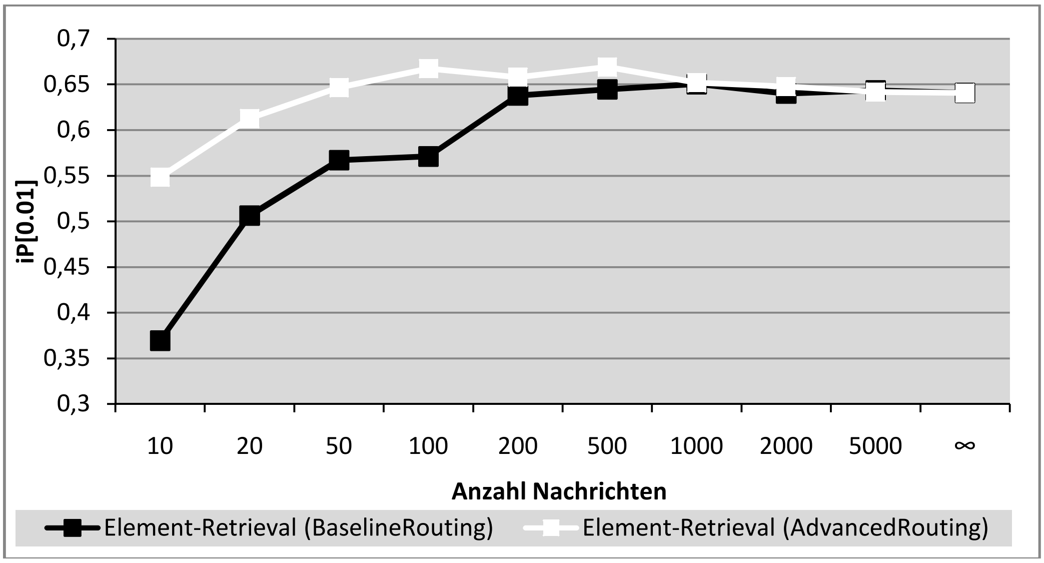


Abbildung 2: Präzision bei unterschiedlicher Anzahl Routingnachrichten beim Baseline- und Advanced-Routing

Es muss also auf die maximale Anzahl unterschiedlicher Teile der DHT zugegriffen werden.

Der gemessenen Präzision gegenübergestellt sind die Anzahl Nachrichten, die zu deren Erreichen jeweils notwendig war. Die Anzahl Nachrichten ergibt sich durch die Anzahl der Peers, die für das Ranking der ausgewählten Postings kontaktiert werden müssen.

Zwei Suchläufe wurden miteinander verglichen, denen unterschiedliche Routingstrategien zugrundeliegen. Beim Baseline-Routing (schwarze Linie) wurden keine speziellen XML-Retrieval Methoden beim Routing der Anfrage und der notwendigen Auswahl von Postings verwendet.

Beim Advanced-Routing (weiße Linie) wurden die im vorigen Abschnitt propagierten XML-Retrieval Techniken verwendet, um beim Routing Postings von den verschiedenen Peers auszuwählen.

Zu sehen ist, dass bis zu einer Anzahl von 1.000 Nachrichten das Advanced-Routing eine höhere Präzision erreicht; bis 500 Nachrichten ist die Differenz dabei signifikant (T-Test). Ab 1000 Nachrichten lässt sich beim Baseline-Routing keine Präzisionsverbesserung mehr erreichen, die Präzision bleibt bei 65% mit geringen Schwankungen. Das Advanced-Routing erreicht bei 500 Nachrichten den Höhepunkt von 66,9% Präzision, danach fällt sie leicht und wie das Baseline-Routing bei ca. 65%.

XML-Retrieval Techniken, beim Routing auf die Wikipediakollektion und die angegebene Anfragemenge angewandt, helfen im Bereich bis 500 bzw. 1000 Nachricht also dabei, die Suchqualität zu erhöhen. Umgekehrt kann durch ihren Einsatz zusätzlich zur Effektivität auch die Effizienz gesteigert werden: Während beim Baseline-Routing mehr als 150 Nachrichten notwendig sind, um eine Präzision von 61% zu erzielen, sind beim Advanced-Routing für den gleichen Präzisionswert lediglich 20 Nachrichten nötig.

Weiterhin ist festzustellen, dass für die Wikipediakollektion und die INEX-Anfragen von 2008 für das beschriebene System, unabhängig vom Routingverfahren, 500 Nachrichten ausreichen, um eine ausreichend hohe Präzision zu erzielen. Es lässt sich keine Verbesserung der Suchqualität durch Hinzunahme weiterer Nachrichten erzielen.

Die erzielte Suchqualität liegt außerdem im Vergleich zu anderen XML-Retrieval Lösungen im oberen Bereich der INEX 2008 Evaluierung. Das beste System (*focused* Task, *ad-hoc* Track) erzielte in 2008 eine Präzision von 69%.

4. Zusammenfassung

In diesem Papier wurde eine XML-Suchmaschine vorgestellt, in der Methoden des verteilten XML-Retrievals implementiert sind. Sie basiert auf einem strukturierten P2P-Netz. Die Indizes für das Speichern von Dokument- und Elementstatistiken sowie der invertierte Index für Postinglisten sind dabei über eine DHT verteilt. Die Evaluierung der Suchmaschine wurde mittels der Wikipediakollektion vorgenommen, die als Testkollektion von INEX fungiert und im XML-Format vorliegt. Dabei konnte gezeigt werden, dass durch Einsatz von XML-Retrieval Techniken die Suchqualität und die Effizienz beim Routing der Anfragen in der DHT verbessert werden kann. Die durchgeführte Evaluierung bezieht sich auf die INEX 2008 Wikipediakollektion. Kürzlich wurde eine neue Version von Wikipedia herausgegeben, die zukünftig zur Evaluierung bei INEX verwendet werden wird. Diese neue Kollektion umfasst 2.666.190 XML-Artikel (50,7 GB) und verfügt über sehr viel semantischere Elementnamen als die Version von 2006. Es wird daher interessant sein zu sehen, wie sich diese neue Wikipediakollektion mit dem hier vorgestellten System durchsuchen lässt.

Referenzen

[Amer-Yahia and Lalmas, 2006] S. Amer-Yahia, Mounia Lalmas. *XML Search: Languages, INEX and Scoring*. SIGMOD RecVol. 35, No. 4, 2006.

[Bray *et al.*, 2006] Tim Bray, Jean, Paoli; C.M. Sperberg-McQueen et al. (eds.). *Extensible Markup Language (XML) 1.1 (Second Edition)*. W3C Recommendation vom 16. August 2006.

[Denoyer and Gallinari, 2006] Ludovic Denoyer, Patrick Gallinari. *The Wikipedia XML Corpus*. In: LNCS, Vol. 4528, Springer-Verlag, 2006.

[El-Ansary and Haridi, 2005] S. El-Ansary, S. Haridi. *An Overview of Structured Overlay Networks.* In: Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks, CRC Press, 2005.

[Luk *et al.*, 2002] R. Luk, H. Leong, T. Dillon, A. Chan. *A Survey in Indexing and Searching XML Documents*. Journal of the American Society for Information Science and Technology, Vol. 53, No. 6, 2002.

[Risson and Moors, 2004] J. Risson, T. Moors. *Survey of research towards robust peer-to-peer networks – search methods.* In: Technical Report UNSW-EE-P2P-1-1, University of NSW, Australia, 2004.

[Robertson *et al.*, 2004] Stephen E. Robertson, Hugo Zaragoza, M. Taylor. *Simple BM25 extension to multiple weighted fields*. In: Proc. of CIKM'04, ACM Press, New York, USA, 2004.

[Steinmetz and Wehrle, 2005] Ralf Steinmetz, Klaus Wehrle (eds.). *Peer-to-Peer Systems and Applications*. Lecture Notes in Computer Science, Vol. 3485, Springer-Verlag, 2005.

[Stoica *et al.*, 2003] Ion Stoica, R. Morris, D. Liben-Nowell, D. Karger, F. Kaashoek, F. Dabek, H. Balakrishnan. *Chord - A Scalable Peer-to-peer Lookup Protocol for Internet Applications*. IEEE/ACM Transactions on Networking, Vol. 11, No. 1, 2003.

[Trotman *et al.*, 2009] Andrew Trotman, Sholo Geva, Jaap Kamps (eds.). *Advances in Focused Retrieval*. Proc. of the 7th International Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2008), LNCS, Springer-Verlag, 2009.

[Winter *et al.*, 2009] Judith Winter, Nikolay Jeliazkov, Gerold Kühne. *Aiming For More Efficiency By Detecting Structural Similarity*. In: Advances in Focused Retrieval, LNCS, Springer-Verlag, 2009.

Probleme und Potenziale bei der Eyetracker-gestützten Evaluierung von interaktivem Retrieval

Matthias Jordan, Sebastian Dungs, Michel Kamsu,
Richard Planert, Tuan Vu Tran, Norbert Fuhr
Universität Duisburg-Essen

Abstract

Um effektives interaktives Retrieval zu erreichen, müssen die Bedienungsoberflächen solcher Systeme in Benutzerevaluationen untersucht werden, wobei das Untersuchungsdesign zu möglichst differenzierten und aussagekräftigen Ergebnissen führen sollte. In diesem Beitrag wird die Evaluierung einer Benutzeroberfläche einer digitalen Bibliothek vorgestellt, wobei zudem eine neue Visualisierungskomponente einbezogen wurde. Wir stellen die Untersuchungsmethodik vor und beschreiben die bei der Durchführung und Auswertung aufgetretenen Probleme. Trotz der ausreichenden Stichprobengröße von 20 Probanden ergaben sich kaum signifikante Unterschiede, doch konnten einige qualitative Aussagen abgeleitet und Folgerungen für zukünftige Untersuchungen dieser Art gezogen werden.

1 Einleitung

Die Benutzerschnittstelle einer Digitalen Bibliothek hat einen großen Einfluss auf den Erfolg einer Suche in der Kollektion. Daher ist es sinnvoll, Benutzerschnittstellen und ihre Komponenten Evaluationen zu unterziehen, um Probleme in der Interaktion zu finden oder auch ihren Beitrag zum Sucherfolg zu ermitteln. Ein oft untersuchtes Problem ist die explorative Suche, bei der das Ziel der Suche unbekannt und die Unsicherheit bei der Suche hoch ist und bei der der Suchende teilweise zahlreiche Wege beschreiten muss, um fündig zu werden.

Kollektionen mit semi-strukturierten Dokumenten ermöglichen dabei Suchpfade, die über die unstrukturierter Dokumente hinausgehen. Sie besitzen neben dem Volltext noch Meta-Informationen, von denen einige Beziehungen zwischen Dokumenten herstellen, die bei der Suche genutzt werden können. Die Nutzung gestaltet sich in herkömmlichen Retrieval-Systemen aber schwierig, da die Interfaces keine direkte Unterstützung dafür anbieten.

DAFFODIL ist eine Metasuchmaschine für Digitale Bibliotheken, die mit semi-strukturierten Dokumenten arbeitet und, neben der Verteilung der Suchanfragen auf mehrere DLs, reichhaltige Werkzeuge zur strategischen Unterstützung anbietet (siehe [Klas, 2007]). Dadurch ist DAFFODIL einerseits ein sehr mächtiges Tool für die Literatursuche; andererseits ist DAFFODIL aber auch komplex und kann für Anfänger unübersichtlich sein.

Um die Beziehungen zwischen Dokument-Attributen zu visualisieren, wurde daher im Rahmen einer Projektgruppe der Universität Duisburg-Essen (siehe [Erlinghagen *et al.*, 2008]) und einer Diplomarbeit (siehe [Tarassenko, 2008]) die Visualisierung „VisAtt“ als Werkzeug im DAFFODIL-Framework entwickelt. Nach Eingabe einer Suchanfrage und Eintreffen der Ergebnisliste visualisiert VisAtt bestimmte Attribute (z.B. Co-Autoren-Beziehungen) in einem Werkzeug neben der Ergebnisliste. Die Hoffnung beim Entwurf der Visualisierung war, dass bestimmte Fragestellungen, die im Lauf einer explorativen Suche auftreten können, leichter in der Visualisierung beantwortet werden können als mit den vorhandenen Werkzeugen.

Die tatsächliche Wirksamkeit der Visualisierung wurde in einer zweiten Projektgruppe, auch mit Hilfe eines Eyetrackers, evaluiert. Ziele dabei waren sowohl Aussagen über die Effektivität der Visualisierung als auch das Identifizieren von Schwierigkeiten bei der Eyetracker-gestützten Evaluation solcher Visualisierungen. Die in dieser Studie gewonnenen Ergebnisse bilden zugleich die wesentliche Grundlage dieses Papers.

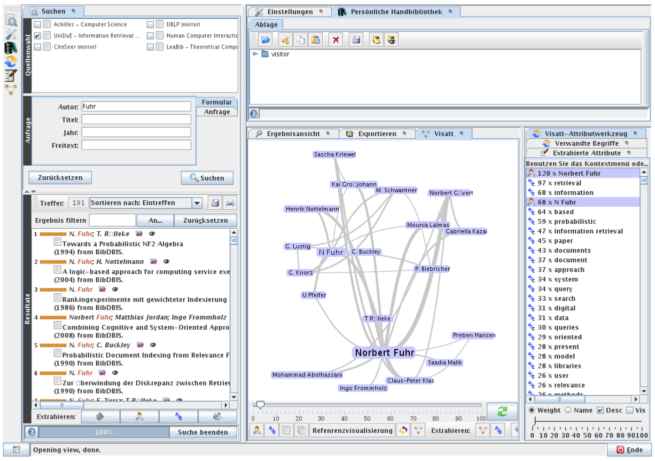
Mittlerweile gibt es eine Reihe von Eyetracker-gestützten Evaluationen von IR-Systemen, z.B. die Studien von Granka et al. und Guan und Cutrell (siehe [Granka *et al.*, 2004] bzw. [Guan and Cutrell, 2007]), die sich jedoch häufig auf die Betrachtung linearer Ergebnislisten beschränken. Daher liegt der Schwerpunkt dieser Arbeit in der Erprobung von Untersuchungstechniken komplexer Visualisierungen; die Ergebnisse der Evaluation selbst sollen nur eine untergeordnete Rolle spielen.

Im Rahmen dieser Arbeit werden wir kurz die Visualisierung „VisAtt“ vorstellen, das Studiendesign beschreiben, die Ergebnisse der Evaluierung diskutieren und abschließend interessante Probleme erörtern, die im Laufe der Durchführung der Studie aufgetreten sind.

2 VisAtt

Die Visualisierung „VisAtt“ ist graphorientiert. Sie visualisiert Attribute als Knoten des Graphen und Beziehungen als Kanten zwischen den Knoten. Die Vielfachheit einer Beziehung (z.B. die Häufigkeit, mit der zwei Autoren gemeinsam Papers der Ergebnisliste verfasst haben) wird in der Dicke einer Kante wiedergegeben. Die Graphen werden initial geordnet, die Knoten können aber vom Benutzer verschoben werden.

In Abbildung 1 ist ein DAFFODIL-Frontend mit VisAtt zu sehen. Im linken Bereich befindet sich das Suchwerkzeug, mit dem der Benutzer seine Suche durchführt und in dem er die Ergebnisliste betrachten kann. Rechts neben der Ergebnisliste befindet sich VisAtt, das in dieser Ansicht die Koautoren-Beziehung der Autoren der Ergebnisliste visualisiert. Neben der Graphen-Visualisierung ist das Attributwerkzeug von VisAtt untergebracht, in dem die Attribute



□ Achilles – Computer Science □ DBLP (mirror)
☑ UniDuE – Information Retrieval ... □ Human Computer Interactio
□ CiteSeer (mirror) □ LeaBib – Theoretical Compu

Autor: Fuhr
Titel:
Jahr:
Freitext:

Treffer: 191 Sortieren nach: Eintreffen
Ergebnis filtern An... Zurücksetzen

1 N. Fuhr; T. R□lleke
Towards a Probabilistic NF2 Algebra (1994) from BibDBIS.
2 N. Fuhr; H. Nottelmann
A logic-based approach for computing service exe (2004) from BibDBIS.
3 N. Fuhr
Rankingexperimente mit gewichteter Indexierung (1986) from BibDBIS.
4 Norbert Fuhr; Matthias Jordan; Ingo Frommholz
Combining Cognitive and System-Oriented Appro (2008) from BibDBIS.
5 N. Fuhr; C. Buckley
Probabilistic Document Indexing from Relevance F (1990) from BibDBIS.
6 N. Fuhr
Zur □berwindung der Diskrepanz zwischen Retriev (1990) from BibDBIS.

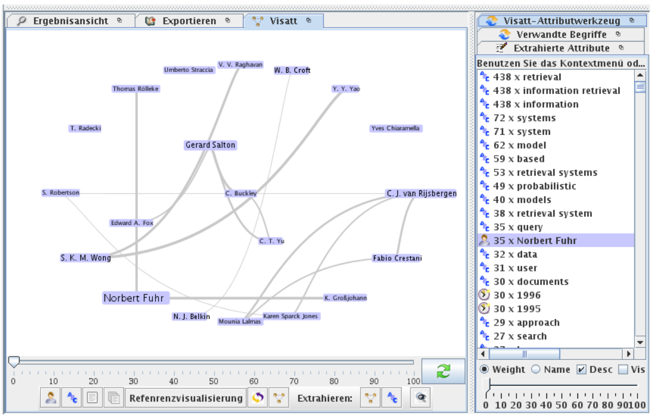
100%

Opening view, done.

Benutzen Sie das Kontextmenü ode...
120 x Norbert Fuhr
97 x retrieval
68 x information
68 x N Fuhr
64 x based
59 x probabilistic
47 x information retrieval
45 x paper
43 x documents
37 x document
37 x approach
34 x system
34 x query
33 x search
31 x digital
31 x data
30 x queries
29 x oriented
28 x present
28 x model
28 x libraries
26 x user
26 x relevance

● Weight ○ Name ☑ Desc □ Vis

Abbildung 1: Screenshot von DAFFODIL mit VisAtt



Benutzen Sie das Kontextmenü od...
438 x retrieval
438 x information retrieval
438 x information
72 x systems
71 x system
62 x model
59 x based
53 x retrieval systems
49 x probabilistic
40 x models
38 x retrieval system
35 x query
35 x Norbert Fuhr
32 x data
31 x user
30 x documents
30 x 1996
30 x 1995
29 x approach
27 x search

● Weight ○ Name ☑ Desc □ Vis

Abbildung 2: Screenshot von VisAtt im Autoren-Netzwerk-Modus

der Ergebnisliste in Listenform angeordnet sind – hier sortiert nach Vorkommenshäufigkeit.

VisAtt erzeugt seine Graphen allein aus den Ergebnissen der jeweils letzten Suche. In Abbildung 2 ist dies beispielhaft für die Autoren-Netzwerk-Visualisierung von VisAtt gezeigt, die nach einer Suche nach „information retrieval“ im Titel-Feld berechnet wurde. Die Knoten des Graphen entsprechen Autoren und die Kanten entsprechen Koautoren-Beziehungen. Die Stärke der Kanten visualisiert die Anzahl der gemeinsamen Veröffentlichungen.

Unter dem Graphen sind einige Einstellungs-Elemente zu sehen; mit ihnen können Cut-Offs für die Kanten-Vielfachheit eingestellt werden (z.B. „zeige nur Kanten mit mehr als 5 Nennungen“). Weitere Buttons ermöglichen das Umschalten in andere Visualisierungs-Modi. Mithilfe dieser Modi sollte es einfacher sein, herauszufinden, wer zentrale Autoren in einem Themenfeld sind, welche Terme häufig und zusammen mit welchen anderen Termen vorkommen, aber auch welche Autoren häufig mit welchen Termen assoziiert sind oder welche Dokumente von welchen Dokumenten zitiert werden.

3 Evaluationen mittels Eyetrackern

Eyetracker bestimmen anhand von Fernmessungen der Pupillen-Ausrichtung die Fixationspunkte eines Benutzers auf einem Objekt – in unserem Fall ein Monitor, auf dem die grafische Benutzeroberfläche von DAFFODIL gezeigt wurde. Die Daten, die ein Eyetracker liefert, sind (vereinfacht) Fixationskoordinaten, die jeweilige Verweildauer auf dieser Koordinate und Zeitstempel. Anhand dieser Daten lassen sich sowohl quantitative als auch qualitative Untersuchungen durchführen. Eine gute Einführung in das Thema bietet Bente in [Bente, 2004].

Für quantitative Untersuchungen werden i.d.R. in der Auswertungssoftware anhand von Screencaptures Bildschirmregionen („Area of Interest“ oder AOI) festgelegt. Fixationen innerhalb dieser AOIs werden dann zu einer Gruppe zusammengefasst, um alle Fixationen innerhalb dieses Bereichs gleich zu behandeln. Es ist z.B. möglich, den Bereich, an dem ein Button dargestellt wird, als AOI zu definieren, um dann alle Fixationen auf diesem Button, unabhängig von der Position innerhalb des Buttons, gesammelt zu betrachten.

Qualitative Untersuchungen werden durch andere Auswertungsmethoden unterstützt; eine davon ist die Beobachtung des Videoprotokolls der Bildschirminteraktion mit den überlagerten Fixationsdaten. Dazu wird während des Experiments ein Video-Screencapture durchgeführt, in dem auch der Mousepointer mit einbezogen wird, und beim Abspielen werden die Fixationen als Punkte auf diesem Film visualisiert. Dadurch lässt sich nachvollziehen, wie der Blickverlauf des Benutzers während des Versuchs war.

4 Studiendesign

Drei Fragen standen bei der Betrachtung von VisAtt im Vordergrund:

1. Hilft VisAtt tatsächlich bei der Literatursuche?
2. Wie hilfreich ist ein Tutorial für die Nutzung der Funktionen von VisAtt?
3. Unterstützt VisAtt bei Known-Item-Suche und bei explorativer Suche?

Zur Untersuchung dieser Fragen wurde die traditionelle userorientierte Evaluierung mit dem Einsatz eines Eyetrackers kombiniert, um nähere Aufschlüsse über Probleme der Visualisierung zu erhalten.

Um mit einer definierten Dokumentmenge arbeiten zu können, wurde ein DAFFODIL-Client auf die Arbeitsgruppen-eigene BibTeX-Literatur-Datenbank eingeschränkt, die zum Zeitpunkt der Untersuchung 4330 Dokumente umfasste.

Der Begriff der „Hilfe“ bzw. „Unterstützung“ wurde in diesem Zusammenhang als positive Beeinflussung von Suchzeit, Sucherfolg, Suchbegriffen und Übersichtsgefühl definiert. Der Sucherfolg wurde gemessen, indem vor Durchführung der Experimente die in der Dokumentkollektion vorhandenen relevanten Dokumente identifiziert und die Ergebnisse der Teilnehmer mit diesen Ergebnissen verglichen wurden.

Als Suchaufgaben wurden zwei Aufgaben zur Known-Item-Suche sowie zwei explorative Suchaufgaben gewählt.

Die Known-Item-Suchen betrafen das Auffinden eines bestimmten Artikels von Nick Belkin über „information seeking“ (im Folgenden „K1“ genannt) und eines Artikels von Norbert Fuhr und Sascha Kriewel über „query formulation“ (K2). Die erste explorative Aufgabe beinhaltete das Finden führender Autoren im Bereich kollaborativer Ansätze (E1). Die zweite Aufgabe zur explorativen Suche hatte zum Inhalt, das Hauptthema gemeinsamer Artikel von Norbert Fuhr und Claus-Peter Klas zu finden (E2).

Variable	Median	Modus
Erfahrung Suchmaschinen	5	4
Häufigkeit Suchmaschinen	7	7
Erfahrung Digitale Bibliotheken	3	1
Häufigkeit Digitale Bibliotheken	3	3
Deutschkenntnisse	$6,5$	7
Englischkenntnisse	5	5

Tabelle 1: Statistische Angaben der Teilnehmer

Zusätzlich sollten die Teilnehmer eine Reihe mit Interface-bezogenen kleinschrittigen Aufgaben bearbeiten. Aufgaben dieses Fragebogens waren z.B. „Wechseln Sie in die Schlüsselwortansicht“ und „Nennen Sie eine Jahreszahl, in der verhältnismäßig viele Dokumente gefunden wurden“. Abschießend sollten die Teilnehmer einen Fragebogen mit Zufriedenheitsmetriken ausfüllen.

Randomisiert wurde die Reihenfolge der Aufgaben und die Zuordnung, welche beiden Aufgaben mit VisAtt zu bearbeiten waren. Um zu prüfen, welchen Einfluss ein Tutorial auf den Erfolg der Suche hat, wurde einer Hälfte der Teilnehmer eine papierbasierte Einführung in VisAtt zur Verfügung gestellt, die sie vor Beginn des Experiments lesen sollten.

Aus den fünf Aufgaben und den zwei Werten der Tutorial-Variablen ergaben sich 10 Gruppen, die mit je zwei Teilnehmern besetzt wurden. Daraus ergaben sich insgesamt 20 Experimente.

Der Versuchsablauf begann mit einer kurzen Einführung in die Untersuchung, die den Zusammenhang und das allgemeine Setting darstellte, aber nicht preisgab, was der Untersuchungsgegenstand war. Die Teilnehmer wurden auch darauf hingewiesen, dass sie die Teilnahme jederzeit ohne Angabe von Gründen abbrechen konnten. Danach wurde den Teilnehmern gruppenabhängig das Tutorial ausgehändigt. An diese Einführung schlossen sich die fünf Aufgaben an; der Fragebogen mit statistischen Fragen wurde den Teilnehmern abschließend vorgelegt.

Während des Experiments arbeiteten die Teilnehmer i.d.R. selbständig. Die einzigen Eingriffe des Versuchsleiters waren Fehlerbehebung in der Applikation, um Softwarekorrektheit zu simulieren, sowie standardisierte Eingriffe zwecks Vorbereitung einzelner Aufgaben. So mußte z.B. zu Beginn einer Suche, die mit VisAtt durchgeführt werden sollte, dem Teilnehmer eine andere Version des DAFFODIL-Clients zugänglich gemacht werden.

An den Experimenten nahmen 10 weibliche und 10 männliche Versuchspersonen teil, 90% von ihnen aus der Altersspanne 20–30 Jahre. Der größte Teil bestand aus Studenten aus den Fachrichtungen Informatik, Kommunikations- und Medienwissenschaften, Bauwesen und Psychologie. Einige statistische Angaben aus Likert-Skalen (1–7 Punkte, 7 ist stärkste Ausprägung) sind in Tabelle 1 zusammengefasst.

5 Ergebnisse

5.1 Unterstützung der Suche

In Tabelle 2 sind die Suchzeiten für die Aufgaben zusammengefasst; die Zeilen mit Suffix „_Vis“ enthalten die Zeiten für die Versuche, in denen VisAtt verwendet wurde. Die Gruppen-Unterschiede sind mit p-Werten oberhalb von 0,2 jeweils nicht signifikant. Ursächlich dafür, dass ein eventuell vorhandener Unterschied nicht mit ausreichender Sicherheit festgestellt werden konnte, ist evtl. das Auftreten von Softwarefehlern, die bei Verwendung von VisAtt die gemessenen Zeiten verfälscht haben.

Aufgabe	# Probanden	Mittelwert	Median
K1_Vis	10	150,90	122,00
K1	10	113,90	100,50
K2_Vis	10	185,20	149,00
K2	10	135,20	129,50
E1_Vis	10	162,90	118,00
E1	10	156,80	172,50
E2_Vis	10	272,80	275,50
E2	10	241,40	200,50

Tabelle 2: Die Bearbeitungszeit (in Sekunden) der einzelnen Aufgaben

Aufgabe	# Probanden	Mittelwert	Median
K1_Vis	10	1,80	2,00
K1	10	2,00	2,00
K2_Vis	10	1,60	2,00
K2	10	1,80	2,00
E1_Vis	10	0,70	0,00
E1	10	0,60	0,00
E2_Vis	10	0,80	0,00
E2	10	1,40	2,00

Tabelle 3: Die Punkteverteilung der einzelnen Aufgaben

Um den Sucherfolg zu messen, wurden den Ergebnissen bei vollständig richtiger Lösung 2 Punkte, bei teilweise richtiger Lösung 1 Punkt und bei falscher Lösung 0 Punkte zugeordnet. Die Ergebnisse sind in Tabelle 3 zusammengefasst. Ein Mann-Whitney-U-Test zeigt, dass die Unterschiede nicht signifikant sind.

Die Suchbegriffe wurden von einigen Teilnehmern anhand von Begriffen aus der Visualisierung geändert. Das kann als Hinweis darauf gewertet werden, dass die Visualisierung u.U. dazu beiträgt, die Suche in neue Richtungen zu leiten, was bei festgefahrenen explorativen Settings hilfreich sein kann.

Ein interessanter quantitativer Zusammenhang konnte zwischen der Erfahrung im Umgang mit digitalen Bibliotheken und der Fixierung von Punkten außerhalb von VisAtt gefunden werden: der Korrelationskoeffizient nach Spearman hierbei war $-0,505$ mit einem $p = 0,023$ – d.h. je mehr Erfahrung ein Benutzer hat, desto weniger Zeit verbringt er außerhalb der Visualisierung (bzw. desto mehr Zeit verbringt er innerhalb). Dies könnte ein Hinweis darauf sein, dass die Interpretation der Visualisierung unerfahrene Benutzer kognitiv tendenziell belastet und erfahrene Benutzer eher bereit sind, sich der Interpretation von VisAtt zu stellen.

Bei der Known-Item-Suche konnte kein Vorteil der VisAtt-Gruppe gegenüber der Kontrollgruppe gefunden werden; VisAtt wurde hier tendenziell weniger beachtet und brachte dort, wo es verwendet wurde, keinen Vorteil. In Abbildung 3 sieht man eine typische Heatmap für diesen Aufgabentyp.

Neben diesen quantitativen Ergebnissen wurde in mehreren Versuchen beobachtet, dass Teilnehmer den Versuchsleiter gebeten haben, eine Aufgabe, die zur Bearbeitung ohne VisAtt vorgesehen war, mit VisAtt durchführen zu dürfen. Das ist ein Hinweis, dass VisAtt benutzerorientierte Qualitäten hat, die ggf. bei einem Nachfolgeversuch ge-

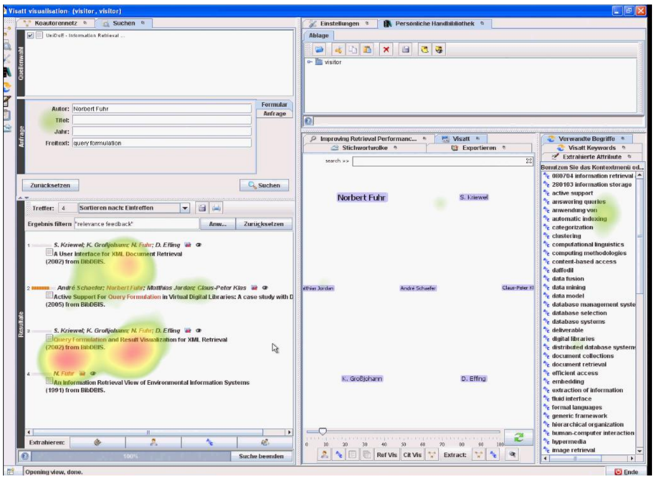


Abbildung 3: charakteristische Heatmap für die Known-Item-Suche

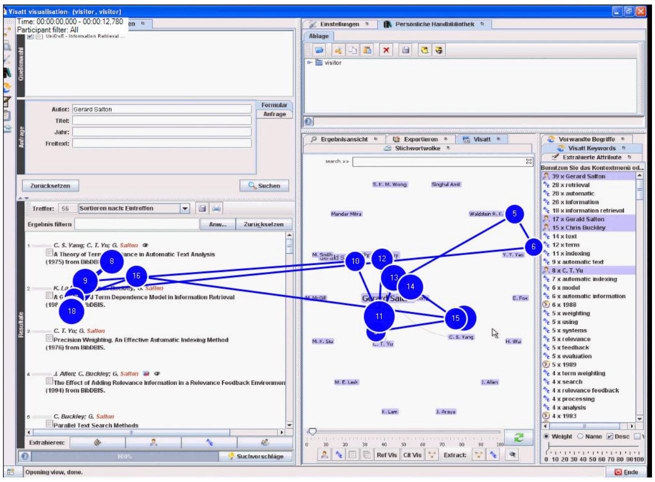


Abbildung 4: Abgleich der Inhalte zwischen Visualisierung und Ergebnismenge

nauer bestimmt werden können. Einen ersten Ansatz dazu bietet die Frage im Abschlussfragebogen nach der subjektiven Beurteilung von VisAtt, die von vielen Teilnehmern im oberen Bereich beantwortet wurde (Median ist 5 auf einer Skala von 1 bis 7).

5.2 Rolle des Tutorials für die Sucherfolge

Die Frage, ob ein Tutorial den Teilnehmern hilft, mit VisAtt bessere Ergebnisse zu erzielen, wurde anhand eines Punktesystems für die Suchergebnisse untersucht. Der Vergleich der Punktzahlen zwischen den Gruppen ohne und mit Tutorial zeigte, dass Teilnehmer ohne Tutorial weniger als zwei Drittel der Punkte der Tutorial-Gruppe erreichten. Die 5-%-Signifikanz ist hier mit $p = 0,058$ mittels Mann-Whitney-U knapp nicht gegeben; eine weitere Untersuchung mit einer größeren Stichprobe sollte hier Aufschluss geben.

Nicht-signifikante Unterschiede zwischen beiden Gruppen gab es auch bei den Ergebnissen der UI-Fragen; hier erreichte die Kontrollgruppe 58% der Punkte der Tutorial-Gruppe mit einem p nach Mann-Whitney-U von $0,067$.

In den Maßen Bearbeitungsdauer der UI-Aufgabe und Gesamtbearbeitungsdauer ließen sich ebenfalls keine signifikanten Unterschiede feststellen. Insgesamt war es also nicht möglich zu zeigen, dass die Unterschiede in den Variablen auf das Tutorial zurückzuführen sind.

5.3 Eigenschaften der Visualisierung

Bei der qualitativen Analyse wurde festgestellt, dass häufig der größte Begriff in VisAtt zuerst fixiert wurde und danach erst der jeweilige Begriff in der Mitte.

Interessant ist, dass bei der Kontrollgruppe ohne Tutorial sehr häufig Blickbewegungen zwischen der Ergebnismenge und VisAtt zu beobachten waren. Dies ist beispielhaft in Abbildung 4 verdeutlicht. Eine mögliche Erklärung dafür ist, dass die Teilnehmer versucht haben, aus dem Vergleich beider Visualisierungen zu schließen, wie die in VisAtt dargestellten Terme zu interpretieren sind. Auch hier standen die von der Schrift her größten Begriffe im Vordergrund der Betrachtung des Benutzers.

5.4 Diskussion

Zusammenfassend lässt sich sagen, dass in dieser Untersuchung ein eventueller Unterschied zwischen der Suchperformance von Benutzern mit und ohne VisAtt nicht ausreichend sicher gezeigt werden konnte. Möglicherweise ist dieser Unterschied vorhanden, aber sehr gering; in diesem Fall stellt sich die Frage, ob die Kosten des Einsatzes dieser Visualisierung (Implementierung, Schulung, Aufmerksamkeit, Arbeitszeit des Benutzers) die Verbesserungen in der Suchperformance aufwiegen.

6 Schwierigkeiten und Ansätze zur Verbesserung des Verfahrens

6.1 Eyetracking

Eine Schwierigkeit, die in der Studie auftrat, war die automatisierte Auswertung der Fixations-Daten. Der Eyetracker ist ein prinzipiell vom zu beobachtenden System (Mensch an Computer mit IIR-Programm) getrenntes System und liefert nur Zeitstempel und Koordinaten, sowie weitere Daten über die Blickbewegung. Anhand dieser Daten ist nicht direkt erkennbar, welcher (logische) Teil des Bildschirms vom Teilnehmer fixiert wurde – es ist nicht einmal direkt ablesbar, ob die Fixation im Fenster des IIR-Programms lag oder nicht. Um dieses Problem zu lösen, arbeitet man, wie oben bereits dargestellt, mit AOIs.

Diese Methode hat die Vorteile, dass sie relativ wenig Arbeitsaufwand für die Definition der AOIs erfordert und die untersuchte Benutzerschnittstelle bzw. dessen Programm als black box behandelt. Der Nachteil ist aber, dass manuelle Tätigkeiten notwendig sind, die ggf. zu Auswertungsfehlern führen können, wenn AOIs nicht exakt definiert sind. Das weitaus größte Problem aber besteht, wenn die vom Benutzer betrachtete Schnittstelle irgendwie gearteten Änderungen unterliegt. Bei Web-orientierten Untersuchungen ist bereits das Scrollen innerhalb einer Seite schwierig zu behandeln. In unserer Untersuchung gab es zwei unterschiedliche Probleme dieser Art. Das erste betrifft Änderungen der Werkzeuge in DAFFODIL: an derselben Stelle, an der VisAtt angezeigt wurde, wurde auch die Detail-Ansicht von Dokumenten in der Ergebnisliste positioniert. Somit ist es alleine anhand von vordefinierten AOIs nicht möglich zwischen Fixationen auf der Detailansicht und solchen auf VisAtt zu unterscheiden. Ähnliche Unschärfen sind aufgrund der variablen Benutzerschnittstelle von DAFFODIL bei prinzipiell allen Werkzeug-Kombinationen denkbar.

Ein weiterer Nachteil dieser Methode ist, dass die AOIs beweglichen Elementen nicht folgen können. Speziell bei der sehr dynamischen Benutzerschnittstelle von VisAtt ist es mit dem herkömmlichen Verfahren nur unter hohem zeitlichen Aufwand möglich, Fixationen auf einzelnen Labels der Visualisierung zu messen.

Um diese Probleme zu umgehen, scheint es aussichtsreich, das zu prüfende Programm so zu ändern, dass automatisch bei Änderungen an der GUI Protokoll über die Grenzen der angezeigten Objekte geführt wird. So könnte z.B. ein Tool, das in den Vordergrund gebracht wird, auslösen, dass die Information über die Grenzen der sichtbaren Flächen der Tools zusammen mit einem Zeitstempel in ein Protokoll geschrieben wird. Dieses Protokoll kann dann mit dem Fixations-Protokoll abgeglichen werden. Ähnliches könnte bei Manipulationen in der Visualisierung erfolgen, womit es dann auch möglich wäre, dynamische Blickbewegungen auf ein einzelnes, durch den Benutzer bewegtes UI-Element, zu verfolgen. Dieser Ansatz wird von uns in einer Nachfolgeversion von DAFFODIL evaluiert werden.

Ein weiteres Problem ist die Vergleichbarkeit der Fixationsdaten zwischen verschiedenen Probanden und/oder Situationen. Um Fragestellungen auf niedriger Ebene besser bearbeiten zu können, scheint es sinnvoll, interessante Situationen zu standardisieren und nicht dem Zufall bzw. der Suchkompetenz der Probanden zu überlassen. Als Beispiel sei hier die Frage genannt, ob innerhalb der Visualisierung das mittlere oder das größte Objekt bevorzugt fixiert wird. Diese Frage kann sogar mithilfe von Screenshots bearbeitet werden, die eine bessere automatisierbare Vergleichbarkeit bieten, als das bei eher zufällig entstandenen Visualisierungs-Konfigurationen der Fall ist. Eine ähnliche Herangehensweise schlagen Käki und Aula mit vordefinierten Anfragen und standardisierten Ergebnislisten vor (siehe [Käki and Aula, 2008]).

6.2 Tutorial

Ein interessanter Zusammenhang besteht zwischen der Selbsteinschätzung der Sprachkompetenz eines Probanden und den Fixationen im VisAtt-Bereich: je höher die Sprachkompetenz, desto mehr Fixationen gibt es dort (Spearman $\rho = 0,633$, $p = 0,003$). Da VisAtt von Sprache keinen starken Gebrauch macht, steht hier zu befürchten, dass bei Nicht-Muttersprachlern das deutschsprachige Tutorial nicht zu besserem Verständnis (und damit intensiverer Nutzung) von VisAtt geführt hat. Eine Lösung wäre, anstatt einer standardisierten Eingabe für den Lernprozess den Ausgang des Lernprozesses zu standardisieren, also z.B. so lange Fragen zuzulassen und zu erklären, bis der jeweilige Proband bestimmte Fragen über VisAtt beantworten kann. Diese Fragen sollten sich allerdings nicht decken mit denen aus der UI-Aufgabe.

Eine andere Methode ist, bei der Auswahl der Probanden ausschließlich auf Muttersprachler zu setzen, da auf diese Weise Einflüsse durch Sprachbarrieren des Interfaces verringert werden können. Eine weitere Beschränkung wäre dabei, ein bestimmtes Maß an Kompetenz in der Sprache vorauszusetzen, in der die Texte in der Kollektion vorliegen. Dadurch könnten Einflüsse durch sprachabhängige Schwierigkeiten bei der Anfrageformulierung reduziert werden.

Weiterhin wäre interessant zu untersuchen, ob es besser ist, die Aufgaben und das Tutorial auf dem Monitor anzuzeigen. In diesem Setting könnten die Blickbewegungen der Teilnehmer auch beim Lesen der Materialien darauf überprüft werden, ob eventuelle Verständnisprobleme vorliegen, bzw. ob Textstellen überhaupt gelesen wurden.

7 Fazit

Die Visualisierung VisAtt ist das Ergebnis einer studentischen Projektgruppe und einer Diplomarbeit. Dennoch konnte in einer umfassenden Evaluation kein signifikanter Vorteil von VisAtt bei Known-Item-Suchen oder explorativen Suchen festgestellt werden. Die einzigen Hinweise auf Verbesserungen liegen im Bereich der Benutzerzufriedenheit. Diese Hinweise korrelieren allerdings nicht mit dem tatsächlichen Sucherfolg und sind teilweise auch nur anekdotisch. Wir kommen daher zu der Auffassung, dass der Entwurf und die Implementierung von erfolgreichen Visualisierungen nicht trivial sind.

Ebenfalls nicht trivial ist die Auswertung von dynamischen Interfaces mittels Eyetracker. Die sich ändernden Positionen von wichtigen Elementen der Visualisierung machen es schwierig, mit herkömmlichen Auswertungsmethoden zu arbeiten. Hier wurde klar, dass mehr Aufwand in die Vorbereitung investiert werden muss. Ob deshalb die Ergebnisse besser (im Sinne von niedrigen p-Werten) werden, ist allerdings noch vollkommen offen.

Zusätzlich zu den Problemen, die durch die Software-Unterstützung des Eyetrackers entstehen, sind sprachabhängige Effekte zu beachten. Dadurch, dass sich in unserem Fall die Landessprache von der Kollektionssprache unterscheidet, müssen Versuchsteilnehmer in zwei Sprachen kompetent sein, um das Rauschen in den Werten gering zu halten und damit die Chance signifikanter Befunde zu erhöhen.

Literatur

[Bente, 2004] Gary Bente. *Lehrbuch Medienpsychologie*, chapter Erfassung und Analyse des Blickverhaltens. Hogrefe, Göttingen, 2004.

[Erlinghagen *et al.*, 2008] André Erlinghagen, Gunnar Meyer, Sergey Tarassenko, Christian Wiacker, and Anne Wolters. Visualisierung von mehrwertigen Attributen. Abschlussbericht der Projektgruppe, 2008.

[Granka *et al.*, 2004] Laura A. Granka, Thorsten Joachims, and Geri Gay. Eye-tracking analysis of user behavior in www search. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 478–479, New York, NY, USA, 2004. ACM.

[Guan and Cutrell, 2007] Zhiwei Guan and Edward Cutrell. An eye tracking study of the effect of target rank on web search. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 417–420, New York, NY, USA, 2007. ACM.

[Käki and Aula, 2008] Mika Käki and Anne Aula. Controlling the complexity in comparing search user interfaces via user studies. *Information Processing and Management*, 44(1):82–91, 2008.

[Klas, 2007] Claus-Peter Klas. *Strategische Unterstützung bei der Informationssuche in Digitalen Bibliotheken.* PhD thesis, University of Duisburg-Essen, 2007.

[Tarassenko, 2008] Sergey Tarassenko. Visualisierung mehrwertiger attribute in digitalen bibliotheken. Diplomarbeit, Universität Duisburg-Essen, 2008.

Relevance Feedback based on Context Information

Felix Engel, Claus-Peter Klas, Matthias Hemmje
FernUniversität in Hagen
Felix.Engel,Claus-Peter.Klas,Matthias.Hemmje@FernUni-Hagen.de

Abstract

In the area of information retrieval the concept of relevance feedback is used to provide high relevant documents to the user. The process of gaining relevance data is usually based on explicit Relevance Feedback. But it turned out, that users are usually not willing to provide such data. This paper describes a Relevance Feedback approach that supports the users with query expansion terms by using implicit Relevance Feedback data.
The Relevance Feedback approach is based on a neural net, which learns the context of a search. The approach is integrated in the DAFFODIL framework. Within DAFFODIL the concept of relevance paths is used for the implementation of an implicit Relevance Feedback mode. The chosen algorithms and data structures indicate a high performance for this approach.

1 Introduction

In this article an approach is described that supports users of an Information Retrieval system (IRs) with query expansion terms. Basis for the expansion term generation is a neural net which learns the context of the user search. Context is defined here as the past and present user concern, which accords to the collected past and current Relevance Feedback data. Those Relevance Feedback data are derived from explicit, implicit or pseudo Relevance Feedback processes.
In order to provide the user with adequate expansion terms, the learn algorithm of the neural net realizes a time dependent adoption of the expansion term output. The time dependent adoption ability of this approach differs from others. Here adoption denotes a highly emphasis of recently learned Relevance Feedback data. The influence of learned patterns gets minor, according to their actuality for the query expansion term generation (or could rather be forgotten). Furthermore the learn algorithm is influenced during the application of implicit Relevance Feedback by learning parameters that depend on the relevance path concept.
The time dependent different impact of Relevance Feedback data on the expansion term generation is realized by a so called *palimpsest* learning rule, which denotes a learning rule that could forget.
As an indicator for the user concern, term co-occurrence states of the Relevance Feedback data are extracted and stored in the neural net. A first performance evaluation has been made.

In the following subsections 1.1, 1.2 and 1.3, a brief introduction is given into relevant topics, concerning the approach of this paper. The section 2 contains a description of the implementation of this approach. In section 3 the results of a first evaluation is presented. Section 4 completes this paper with a summary of the Relevance Feedback approach and an outlook for future works.

1.1 Relevance Feedback

Relevance Feedback is a technique that is widely applied in Information Retrieval (IR). It aims to optimize the search result of an IR process. Relevance Feedback is based on the assumption, that a retrieval result could be optimized by relevance judgments over search results by the IRs user. With these user given relevance judgments it is possible to calculate query expansion terms. Three distinct Relevance Feedback modes are discussed in the literature:

Explicit Relevance Feedback The user marks a search result explicit as relevant according to his query.

Implicit Relevance Feedback On basis of the interaction between the user and the retrieval system a conclusion is drawn, if a search result is relevant to a user query or not.

Pseudo Relevance Feedback On basis of the assumption, that the ranking of the search system is equal to the relevance perception of the user, the first n search results are taken for a Relevance Feedback process.

Google e.g. uses a Relevance Feedback mechanism that is triggered through their *related* function, which in turn could be invoked through a link besides a search result.

1.2 Neural Networks

Neural net approaches are applied in various domains. E.g. in the industry neuronal nets are used for quality assurance, optimization of processes or image processing.
In IR neuronal nets are applied for supporting retrieval systems or could even serve as basis for the whole retrieval process.
Various IR applications exist that base on neuronal nets, like e.g. described in [Bordogna *et al.*, 1996]. *Bordogna* describes a similar approach to the one in this article. She uses likely to this approach a *Hopfield* like neuronal net that calculates query expansion terms on basis of Relevance Feedback data. The most significant difference to this approach is in the time dependent output of this approach.
To sum it up short: a neural net is an information processing system that consists of a set of connected arithmetic units

(neurons). This net of connected arithmetic units has the ability to learn certain states which are used to calculate an output to a given query. A learning rule is needed in order to teach the net with certain states, as well as an activation rule that activates the net in order to get the output data.
Various net types exist, which differ for instance in topology, learn rule, activation rule and the information flow. In this approach a Spreading Activation Net is used, which is described in the subsection 2.1.

1.3 Term co-occurrence

Term co-occurrence denominates a measurement for the togetherness of two terms that appear in a document. The idea of term co-occurrence is, that if two terms exist together in one document, they have a more or less strong relation. *Peat* et al. references in [Peat and Willett, 1990] a statement of *van Rijsbergen* who describes the idea of term co-occurrence: ”*If an index term is good at discriminating relevant from nonrelevant documents then any closely associated index term is also likely to be good at this*”. Term associations could therefore be used to find further terms with a similar distinction ability between relevant and non relevant documents. This assumption assumes that the used query terms are good discriminators.
A disadvantage of the term co-occurrence is, that terms that appear frequently in texts are often not very significant for it. Those terms are e.g. articles. Suitable to this disadvantage *Efthimiadis* states in [Efthimiadis, 1990] that especially terms with a middle frequency in a text, could characterize a text at most (therefore they are significant). This fact is considered in the learning rule with continuous neuron activation states (see paragraph *Learning Rule*). The term co-occurrence in Relevance Feedback data is used here to get a depiction of the users search context into the neural net.

2 Implementation

The approach described in this article exploits the applicability of a Spreading Activation Net to learn and reflect the intensity of co-occurrence states of given (textual) Relevance Feedback data.
It is argued here that the frequency of a term co-occurrence is a clue for their importance to the actual search. The term co-occurrence states are therefore depicted through a learn process into the net weight matrix.
Therefore the approach in this article is to train the Spreading Activation Net with term co-occurrence states via a learning rule. In this case the well known *Hebb* learning rule was used. Some adoptions to this rule were made to model the demand of slowly forgetting learned patterns (see paragraph *Learning Rule*) in order to prefer actual Relevance Feedback data during the net activation (see paragraph 2.1).
Because of the Spreading Activation Net character, just working with a fixed number of neurons, a fixed term space (see paragraph *Term Space*) is used. Therefore this approach is limited until now to more or less static data collections.
For the evaluation of this approach a term space consisting of around 27 500 terms, extracted from the first 10 000 abstracts of the *CiteSeer* database is used.
The following subsections will describe in more detail all the named aspects in context with the query expansion calculation.

2.1 Spreading Activation Net

A *Spreading Activation Net* denotes a special type of a neural net, which is in some aspects related to the *Hopfield* net. The main difference between both net types consists in the absence of an energy function in the Spreading Activation Net.
A Spreading Activation Net could be considered as a graph with nodes and edges. The edges (rather connections) between the nodes (rather neurons) are symmetric weighted. Those neurons are fully connected, but irreflexive. The strength of the connections between the neurons are calculated by a learning rule, like e.g. the *Hebb* learning rule. A squared and symmetric weight matrix $w_{ij} \in W$, with zero values in the diagonal is then used to represent the connection strength. The values in this matrix are calculated by the frequency of the co-occurrence of two terms by means of the *Hebb* rule.
Furthermore the neurons of the net have an activation state that is equivalent to the value of the corresponding *pattern vector* element $x_i \in p$ (see paragraph *Neuron Model*). The values of those activation states could be either binary or continuous. Figure 1 shows an example of a simple Spreading Activation net with associated weight matrix.

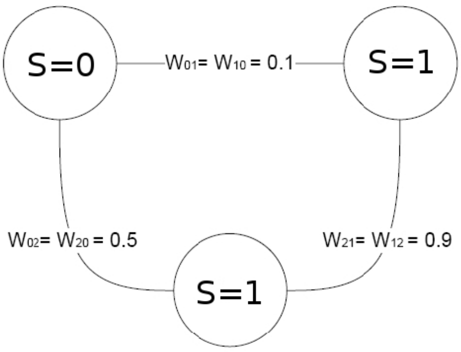


Figure 1: Spreading Activation Net

$$\mathbf{W} = \begin{pmatrix} 0 & 0.1 & 0.5 \\ 0.1 & 0 & 0.9 \\ 0.5 & 0.9 & 0 \end{pmatrix}$$

Weight matrix for figure 1

This net has to be instantiated with a fix amount of neurons, that could not be modified after the instantiation.
Different IR approaches exist that bases on Spreading Activation Net types (see [Kwok, 1989], [Chen *et al.*, 1993]), whereby the application of the net is manifold. The denomination ”spreading activation” is derived from the net feature to spread activation from a few activated neurons through the net. An activation rule is responsible for the neuron activation and therefore for the net output.

Neuron Model In order to generate query expansion terms on basis of a Spreading Activation Net, a depiction is needed from the terms in the term space (see paragraph *Term space*) to the neurons of the Spreading Activation Net. E.g. in the example net in figure 1 the term space consists only of three terms, that have to be depicted to the three neurons of the corresponding net. This depiction is done by an explicit allocation of one term to one neuron of the net. The allocation is implemented by the elements x_i of a pattern vector p, in which each position (or index) is allocated with one term of the term space. The cardinality of p is thereby equivalent to the number of the net neurons (and hence equivalent to the number of terms in the term space). This means the activation state of a neuron is encoded through setting the value of an element of the pattern vector (which is allocated to a term in the term space). There are two different pattern vector element values analyzed:

- Binary pattern elements. Those relate to a binary neuron activation degree (set{0,1} is used).
- Continuous pattern elements. Those relate to a continuous neuron activation degree (interval [0,1] is used).

Learning Rule For the net training with co-occurrence states of Relevance Feedback data, a modified *Hebb* learning rule (or outer product rule) is applied. The origin of this rule traces back to the publication [Hebb, 1949] of the psychologist *D. Hebb*. He stated the assumption, in turn, that if two cells are active at the same time, their connection weight should be strengthened. This assumption reflects exactly the learning of term co-occurrence states, whereas the net should strengthen the connection weight of two neurons if two terms occur at the same time in a presented text. This could be mathematically formalized through the equation 1.

$$\Delta w_{ij} = \gamma x_i x_j \tag{1}$$

Where Δw_{ij} reflects the difference between the association strength of two neurons x_i and x_j in time t_m and t_{m-1}. γ is a learning constant. The *Hebb* learning rule learns unsupervised and could be applied incremental or in batch mode.

In general the state of a neuron x is binary $\{-1, 1\}$, whereas some approaches use the values {0,1}. Some performance optimization reasons lead to the usage of the values {0,1} in this approach. Reason for this decision was that the patterns that should be learned, as well as the patterns that are used for activating the net are sparsely encoded. Firstly, there exists special data structures to store these kind of data efficiently (zero values causes no memory allocation). Secondly, the value of a weighted connection between neurons changes only if one of the activation degrees is unequal zero.

The first adaption to the rule aims to model forgetting. *Hopfield* proposes in [Hopfield, 1982] a limitation of the weight growing, in order to model forgetting. A different approach is found in [Storkey and Valabregue, 1999] and [Kimoto and Okada, 2000], where forgetting is modeled through a proportionate small decay of weights in each learn cycle.

In this approach both suggestions are used. The limitation of growth prevents excessive growths of connection strength. A small decay of weight in each learn cycle models the decrease of interest in neuron connections. So far the *Hebb* learning rule is extended to the following equation (2).

$$w_{ij}(t) = w_{ij}(t-1) + \gamma x_i x_j - decayRate \tag{2}$$

Whereby $0 \leq w_{ij}(t) \leq |growingBorder|$.

Considering the problem of term frequencies and the resulting problems for term co-occurrence, that is mentioned in the subsection 1.3, two different learn approaches are implemented:

The first approach uses binary activation states. That means if a term exists in a given Relevance Feedback data, the allocated neuron is either active (activation state is set to 1) or not active (activation state is set to 0).

In a second approach the activation state of a neuron is continuous and depends on the descriptive quality of the term to characterize a text. As *Efthimiadis* stated in [Efthimiadis, 1990] terms with a middle text frequency are those that characterizes a text at most. In contrast terms with a high or low text frequency (e.g. articles) characterize a text less. On basis of this awareness in the second approach the terms with a middle term frequency are emphasised during the learn process.

To get a proper depiction from term frequency to an appropriate neuron activation degree, a function is needed that realizes the following conditions:

- the function course has to be bell shaped
- the function maximum has to be right above the middle term frequency
- the function maximum is $f(\mu) = 1$, whereby μ is the middle term frequency
- the width of the function course should depend on the term variance σ
- the intersection with the y-axis should be constant for each term distribution

All of the mentioned properties are found in equation 3.

$$f(x) = e^{-k((x-\mu)/\sigma)} \tag{3}$$

Whereby the parameter k is responsible for the height of intersection from the function course and y-axis.

The parameter k has to be calculated for each term distribution. The middle term frequency μ is set to the meridian of the term distribution.

The continuous activation degrees should cause a low connection weight if both terms have a low or high term frequency. High activation degrees are caused through a middle activation degree of one or both terms.

A high connection weight causes in the activation process in a higher activation degree of a neuron. Therefore it should be higher ranked in the output of the net.

Activation Rule and Output After the net has learned one or more patterns, it can be activated. In order to start an activation process the terms (neurons) that should be activated have to be encoded into a pattern vector *p* (see section 2.1). This pattern vector indicates then which neurons should be initially activated. If some neurons are activated, the activation function is used to spread the activation through the whole net.

In this approach a *sigmoidal* function is used in the activation process. As a specification of the function the *tanh* function is used, which depicts the function input into the interval [-1,1]. Whereby, through the restriction of the learn rule the input is always positive. The function input is calculated through the sum of products of upstream neuron activations with their connection weights. The whole activation process is formalized through equation 4, according to the activation approach of *Chen* in [Chen *et al.*, 1993].

$$x_j(t+1) = f_{tanh}(\sum_{i=1}^{N} w_{ij} * x_i(t)) \tag{4}$$

The parameter N identifies the number of used neurons. The parameter x is equivalent to the activation degree of the upstream neurons.

The calculation of the activation degree is applied sequentially and in random order for each neuron based on the initially activated neurons. The stop criterion is thereby either a fix number of activation passes or no changes in the neuron activation states.

During the activation process the activation spreads through the net and causes a high activation state of neurons with a strong connection to the initial activated neurons. In this approach the net output is a ranked set (according to the activation degree) of query expansions terms.

Term Space The Spreading Activation Net is not flexible in the way that it could add or delete neurons after the initiation. While each neuron of the net represents a possible query expansion term a fixed term space is needed. This fact might not be a problem for digital libraries, but it prevents the generalization of this approach to a more dynamic system like e.g. the internet.

For the evaluation the first 10 000 abstracts of the *CiteSeer* database are used. The terms of those abstracts were firstly cleaned from stopwords and secondly reduced to their word stem.

The word stem reduction is done because of two reasons. Firstly, it reduces the number of used neurons and this helps to save memory space. Secondly, it has to be taken into account that each term of the term space is exactly allocated to one neuron of the Spreading Activation Net. If the terms of the term space are not reduced to their word stem, problems or rather blur with term flexions occur.

For example if the net should be activated with the term *"effective"*, only the neuron that is allocated to the term *"effective"* would be activated. The neurons that are allocated to the terms *"effectiveness"*, *"effectively"* or "'*effecting*"' would not be activated, even if they denote the same concept.

The disadvantage of the word stem reduction is that terms with different meaning could have the same word stem. The word stem *"gener"* for example is the same for the terms *"generalized"* and *"generators"*, which identifies definitively two totally different concepts. A further disadvantage of the word stem reduction is that word stems seem sometimes cryptic. As an example the term *"beings"* would be reduced to the stem *"be"*.

2.2 Integration into the Daffodil project

The described Relevance Feedback approach of this paper has been successfully implemented and integrated in the *Daffodil*-Framework. It includes three Relevance Feedback modes – explicit, implicit and pseudo – as well as a task based search support.

The implementation of implicit Relevance Feedback and the support of task based searching is described in the following paragraphs.

Daffodil The DAFFODIL-System is an experimental system for IR and collaborative services in the field of higher education for the domain of computer science and others. DAFFODIL is a virtual digital library system targeted at strategic support of users during the information seeking and retrieval process ([Fuhr *et al.*, 2002]). It provides basic and high-level search functions for exploring and managing digital library objects including meta-data annotations over a federation of heterogeneous digital libraries (DLs) (see [Kriewel *et al.*, 2004] for a function overview). For structuring the functionality, the concept of high-level search activities for strategic support as proposed by [Bates, 1979] are employed, and in this way provide functionality beyond today's DL. A comprehensive evaluation in [Klas *et al.*, 2004] showed that the system supported enables most of the information seeking and retrieval aspects needed for a computer scientist daily work.

Implicit Relevance Feedback The great challenge of implementing an implicit Relevance Feedback support is due to the problem of finding which search result is really relevant to a user query. Only those relevant search results should trigger a Relevance Feedback process. The problem in this case is how to find a significant trigger that is crucial for a learning process. Likewise it has to be taken into account, that even if multiple triggers exist not every trigger has the same importance.

The solution to the first problem is found in the relevance path concept that is described in [White, 2004]. Thesis of the relevance path concept is, that as further a user walks along a relevance path, the more relevant the search result is. The relevance path consists of stations and connections between those stations. Whereby a station in this sense denominates a representation form of a search result.

The solution to the second mentioned problem is the usage of a learn constant γ. Such a learn constant should affect the influence of a learn process on the connection strength of neurons.

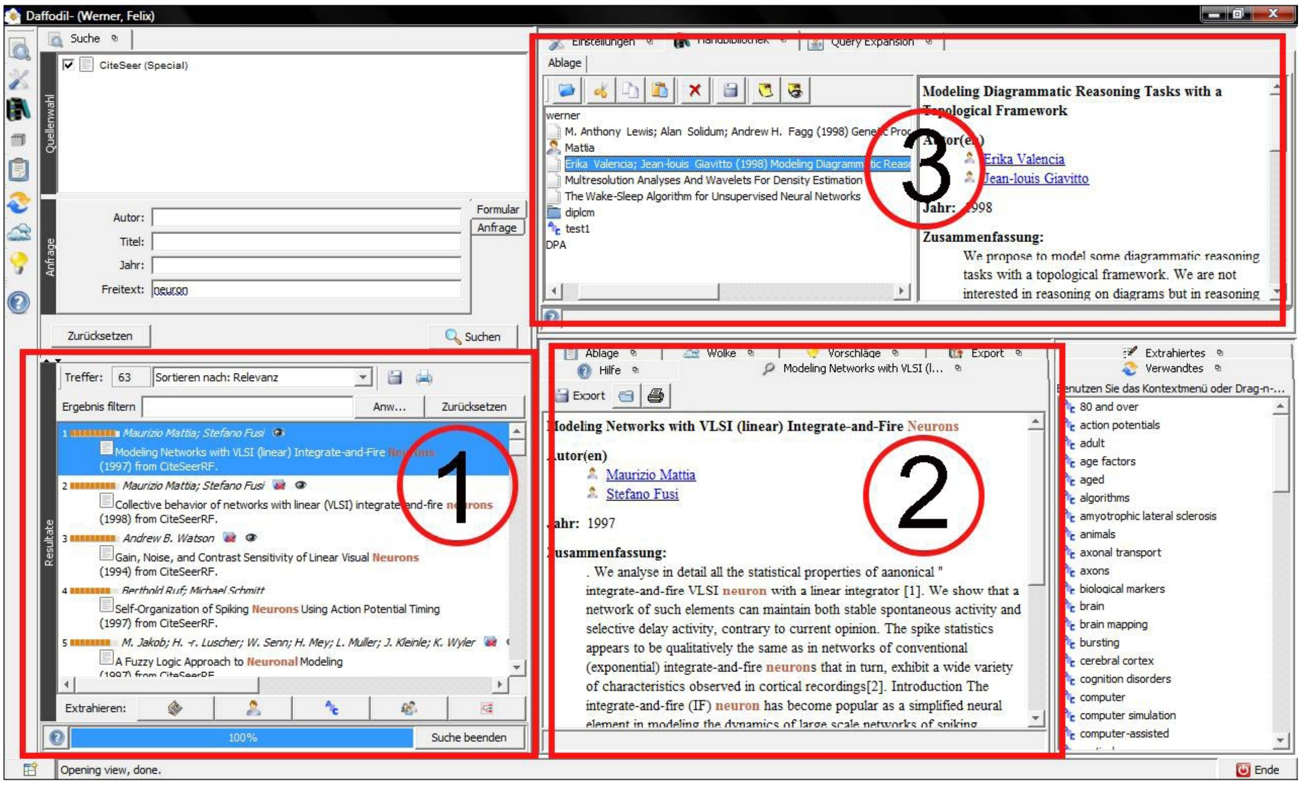
In DAFFODIL a first relevance path station or representation of a search result could be the *result list*. The *result list* prepares all search results to a query and presents it to the user in form of the attributes title, authors names, year and the name of the library where it was found.

If a search result is then clicked, the *detail view* appears and shows additional information, like the abstract of the found search result. Therefore the *detail view* is a second example for a relevance station on the path. According to the relevance path concept the learning process on this relevance station should have a higher impact on the learn process.

Nevertheless, only the representation of a search result in the *detail view* is no indicator for the real relevance of this search result. Because the user could decide to reject the search result as irrelevant after reading the details. For this reason the learn parameter is however set in this case to zero. If the user initializes an export of a search result from the *result list*, it could be assumed that the user has knowledge about this search result, so it might be relevant for this search. In this case the learning parameter is set unequal zero.

An example starting station of a relevance path in *Daffodil* could be the *result list*. The *detail view* could then be the second station of the relevance path. As a next relevance station the Personal Library PLib could be seen. In the PLib The user could store any found object in the personal library. Therefore the objects in the PLib could be seen as high relevant objects. The described relevance path is depicted in the figure 2.

On the basis of this example a relevance path could have the following learning parameters proposed in table 1. The allocation of the proposed parameters for relevance stations and actions are not formalized yet. It just reflects the different importance of an action executed on various relevance



☑ CiteSeer (Special)

Autor:
Titel:
Jahr:
Freitext: neuron

Treffer: 63 Sortieren nach: Relevanz

Ergebnis filtern

1 *Maurizio Mattia; Stefano Fusi*
Modeling Networks with VLSI (linear) Integrate-and-Fire Neurons
(1997) from CiteSeerRF.

2 *Maurizio Mattia; Stefano Fusi*
Collective behavior of networks with linear (VLSI) integrate-and-fire neurons
(1998) from CiteSeerRF.

3 *Andrew B. Watson*
Gain, Noise, and Contrast Sensitivity of Linear Visual Neurons
(1994) from CiteSeerRF.

4 *Berthold Ruf; Michael Schmitt*
Self-Organization of Spiking Neurons Using Action Potential Timing
(1997) from CiteSeerRF.

5 *M. Jakob; H. -r. Luscher; W. Senn; H. Mey; L. Muller; J. Kleinle; K. Wyler*
A Fuzzy Logic Approach to Neuronal Modeling
(1997) from CiteSeerRF.

Opening view, done.

werner
M. Anthony Lewis; Alan Solidum; Andrew H. Fagg (1998) Genetic Proc
Mattia
Erika Valencia; Jean-louis Giavitto (1998) Modeling Diagrammatic Reas
Multresolution Analyses And Wavelets For Density Estimation
The Wake-Sleep Algorithm for Unsupervised Neural Networks
diplcm
test1
DPA

Modeling Diagrammatic Reasoning Tasks with a Topological Framework

Autor(en)
Erika Valencia
Jean-louis Giavitto

Jahr: 1998

Zusammenfassung:
We propose to model some diagrammatic reasoning tasks with a topological framework. We are not interested in reasoning on diagrams but in reasoning

Modeling Networks with VLSI (l...

Modeling Networks with VLSI (linear) Integrate-and-Fire Neurons

Autor(en)
Maurizio Mattia
Stefano Fusi

Jahr: 1997

Zusammenfassung:
. We analyse in detail all the statistical properties of aanonical " integrate-and-fire VLSI neuron with a linear integrator [1]. We show that a network of such elements can maintain both stable spontaneous activity and selective delay activity, contrary to current opinion. The spike statistics appears to be qualitatively the same as in networks of conventional (exponential) integrate-and-fire neurons that in turn, exhibit a wide variety of characteristics observed in cortical recordings[2]. Introduction The integrate-and-fire (IF) neuron has become popular as a simplified neural element in modeling the dynamics of large scale networks of spiking

Benutzen Sie das Kontextmenü oder Drag-n-...
80 and over
action potentials
adult
age factors
aged
algorithms
amyotrophic lateral sclerosis
animals
axonal transport
axons
biological markers
brain
brain mapping
bursting
cerebral cortex
cognition disorders
computer
computer simulation
computer-assisted

Figure 2: Example Relevance Path in *Daffodil*

stations.

station	action	learn parameter
1. Result list	save in PLib	1
	export	0.33
2. Detail view	add term to search formula	0.66
	save	0.66
	clipboard	0.66
3. Personal library	user tags	0.7
	delete	-0.7
	clipboard	0.99
	export	0.99

Table 1: Learn parameter of relevance stations

Example By means of an example the functionality of the learning approach on the weight matrix, as well as the effect of the relevance path concept in the implicit Relevance Feedback mode, should be illustrated.
Assuming the term space of this example consists of the ten, stemmed terms: *co:=collect, de:=deriv, la:=languag, li:=linguist, na:=nature, pa:=paper, pre:=present, pro:=problem, th:=theoret* and *an:=analysi*. Whereby the terms *la, na* and *pro* are included in the Relevance Feedback data (document abstracts) *a, b, c, d* according to the following list:

- *la* included in abstract *a, b*
- *na* included in abstract *a, c, d*
- *pro* included in abstract *a, d, b*

Furthermore the decay rate of the learning rule is set to 0.25, and the growing border to 2. The learning constant γ is set according to table 1.
Some search and working actions were then executed and causes modifications of the neuronal connection weights in the weight matrix. Those actions modifications are illustrated in the table 2. Initially all connection weights are set to zero.

Task Based Relevance Feedback Task based learning means the integration of multiple search sessions into one working task. A working task could be in this sense the work on e.g. a master thesis, lecture or seminar theme. To get a closer depiction of the users working context and to reach a better query expansion support it is an overvalue, if just the Relevance Feedback data of the task the user is working in, trains the net. Relevance Feedback data that depends to a different or neither task (the global task), could blur the expansion result.
Task based learning is realized through training of different nets. This means that each task has its own net. Additionally a overall (or global) net learns every Relevance Feedback data (if not explicitly switched off).

Contextual User Model based on *Daffodil* All activities of the user, when performing a search task with the system, are logged. Along with the activities the time stamp, all user entered information and all digital objects, e.g. document metadata, full text links, queries, authors, terms, journals, etc, are stored. These activities are divided in ten high-level categories, depicted in table 3. For a deeper description of these categories see [Klas *et al.*, 2006].
The described events form a path of events specific to a certain task. With the help of the in DAFFODIL integrated task tool, the system can even train over sessions and provide in this way sophisticated and specific recommendations for a

Action	Connection weight
Reading the details of Abstract *a, b, c, d*	—
Add terms *pro* and *la* to search formula	$w_{pr,la} = 0.66$
Add the abstracts *c, d, b, a* in the PLib (in the specified order)	$w_{na,la} = 1.0; w_{pro,la} = 1.91; w_{na,pro} = 1.5$
Copy *d* from the PLib to the clipboard	$w_{na,la} = 0.75; w_{pro,la} = 1.66; w_{na,pro} = 2.0$
Delete abstract *b*	$w_{na,la} = 0.5; w_{pro,la} = 0.41; w_{na,pro} = 1.75$
Add abstract *e* to the PLib	$w_{na,la} = 0.25; w_{pro,la} = 0.16; w_{na,pro} = 1.5$

Table 2: Table of actions and connection weights

Search	any event of query formulation or filter conditions along with the search sources and system response
Navigate	any event of selecting a specific item link or following link
Inspect	any event accessing or inspecting a object
Display	visualization events, e.g. resorting up to visualizations in tag clouds or clusters.
Browse	events changing the view point without changing the visualization, e.g. scrolling or using sliders to zoom.
Store	events which create a permanent or temporary copy of an object, e.g. printing, clip board storage or the personal library of DAFFODIL.
Annotate	any event that adds additional information to an existing document, e.g. ratings, comments or tags
Author	events on creating new information, e.g. writing a paper within the DL system
Help	any event, where the system provides help to the user either on requests or automatically.
Communicate	any event, when a user communicates with other users, e.g. sharing objects or ask an information professional for help.

Table 3: Logged events in DAFFODIL

task.
The overall contextual user model within the DAFFODIL framework consists of three elements:

Task The user either chooses no tasks (global) or specifies a task. This way, the system can do task specific learning over search sessions and provide specific recommendations.

Search path The complete search path, related to a task as described above.

Information objects All seen, unseen and touched information objects related to a task.

With this context data it is possible to better understand the user on the information behavior in an information seeking and search task. It of course does not excuse from running a real user evaluation.
The logged information was already used for adaptive recommendation services described in [Klas *et al.*, 2008]. Furthermore the logging facility was used in the INEX project ([Malik *et al.*, 2006]).

3 Evaluation

The implementation of this Relevance Feedback approach is only evaluated in regard to its performance. An evaluation that addresses the quality of returned expansion is not yet done.

3.1 Performance Evaluation

In order to test the performance of the implementation of this approach some technical aspects are proven. Those are in detail:

- **length** of abstract: number of terms
- Time to **learn** a pattern (in milliseconds)
- number of **connections**: the number of connections between neurons unequal zero
- **activation**: time to activate the net (in milliseconds)
- number of **cycles**: how much cycles are needed in order to reach a net output

All of those aspects are proved after an explicit Relevance Feedback pass. In order to get a neutral evaluation result the parameters are set to the following:

- Learn parameter is set to 1.
- The growing border is set to 2.
- The decay rate is set to 0.25 (eight learn cycles without a repeated term co-occurrence are needed in order to completely forget a connection)

In this evaluation phase explicit Relevance Feedback was used to the first 20 search results according to the query term ”neuron”. The collected data in this evaluation step can be seen in table 4.

length	learn	connections	activation	cycles
84	15	1847	78	7
89	94	4138	62	9
81	31	5173	62	9
96	31	7363	78	9
95	47	8554	78	9
79	47	8345	78	9
103	47	9489	94	10
87	47	10221	94	9
88	46	9765	78	8
94	47	9734	78	9
103	46	10913	78	9
84	62	10560	93	9
86	47	10337	78	9
95	62	11139	94	9
92	47	11195	94	9
82	47	11906	93	9
82	47	12138	93	9
86	46	12096	93	9
84	31	11236	78	9
98	47	10837	78	10

Table 4: Data from evaluation two

Resume A resume of the important results can be given as following:

- The number of terms in the learned abstracts averages in this situation is 88,52.
- The time to learn the net weights averages 48,8 milliseconds.
- The activation process averages 80,72 milliseconds

Interesting is the attitude at the 9th learn step. Here the numbers of connection unequal zero decrease for the first time. During the learn steps nine to twenty the number of connection swings into the range between 9734 and 12138 connections. The reason for this is the decay rate which destroys connection after the 8th learn cycle.

4 Summary and Outlook

In this paper an approach for context based query expansion term generation is proposed. This approach bases on a Spreading Activation neural net, that learns by means of past and recent Relevance Feedback data the actual situation of the user search in a given task, to provide him with contextualized related query terms. Hereby the concept of term co-occurrence is used to map relevant search results (Relevance Feedback data) into the neural net.

An efficient implementation and a performance evaluation showed a promising behavior in order to investigate this approach with user evaluations.

Additional features of this approach are depicted in order to show the flexibility of this approach. This is shown with the application of task based searching and implicit Relevance Feedback.

Further research should be done in the following directions:

Implementation of Relevance Feedback Services Some services should be set up within the DAFFODIL framework which gathers and delivers the context user model. Along with those services further Relevance Feedback algorithms could be implemented and evaluated against each other.

System oriented evaluation Evaluation series should be done in order to check the Relevance Feedback performance based the proposed algorithm. Standard test beds like *Trec* should be used for it.

DAFFODIL system evaluation Some DAFFODIL internal evaluation should be done by creating a number of DAFFODIL searches. The relevant results of these searches will be stored in the PLib. Along with the logged path data, investigations should be made first in the neural net algorithm for relevant terms. But the more interesting case is to reflect the chosen weights (see table 1) of the path stations with respect to the found documents, e.g. which action of the user has the highest impact on relevant documents.

User evaluation A user evaluation should be made to get qualitative user opinions on the suggested terms.

Enhancement of the implicit Relevance Feedback data Based on the logging system within the DAFFODIL framework the implicit Relevance Feedback data should be enhanced by the user data of the logging system, in order to get a more exact depiction of the user search context.

Collaboration The approach could be extended to support collaborative work. Therefore the net access has to be prepared, so that a group or a team could work at same time with the same net.
Another way, in this sense, could be to share a net with other users on a similar task. Some other applications are possible in the case of sharing nets, e.g. a consolidation of two or more nets could be realized through a simple matrix addition.

Open term space An extension of the applicability of this approach could be reached if the neural net could be released from the fixed amount of neurons, which have to be initially instantiated here. The here used Spreading Activation net does not have the ability to decrease or increase the amount of neurons after the instantiation. The adjustment of the net type to an approach with the possibility to change the amount of neurons, could extend the applicability of this approach to dynamic systems like e.g. the internet.

References

[Bates, 1979] M. J. Bates. Information search tactics. *Journal of the American Society for Information Science*, 30(4):205–214, 1979.

[Bordogna *et al.*, 1996] G Bordogna, G. Pasi, and A. Petrosino. Relevance feedback based on a neural network. *EUFIT '96*, 2:846–850, 1996.

[Chen *et al.*, 1993] H. Chen, K. J. Lynch, and K. Basu. Generating, integrating, and activating thesauri for concept-based document retrieval. *IEEE Expert*, 8(2):25–34, 1993.

[Efthimiadis, 1990] E. N. Efthimiadis. User choices: A new yardstick for the evaluation of ranking algorithms for interactive query expansion. *Information Processing and Management*, 31(4):605–620, 1990.

[Fuhr *et al.*, 2002] Norbert Fuhr, Claus-Peter Klas, André Schaefer, and Peter Mutschke. Daffodil: An integrated desktop for supporting high-level search activities in federated digital libraries. In *Research and Advanced Technology for Digital Libraries. 6th European Conference, ECDL 2002*, pages 597–612, Heidelberg et al., 2002. Springer.

[Hebb, 1949] D. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. Psychology Press, 1949.

[Hopfield, 1982] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *PNAS (Proceeding of the National Academy of Science)*, 79:2554–2558, 1982.

[Kimoto and Okada, 2000] T. Kimoto and M. Okada. Sparsely encoded associative memory model with forgetting process. *IEICE Trans Inf Syst*, E85-D:1938–1945, 2000.

[Klas *et al.*, 2004] Claus-Peter Klas, Norbert Fuhr, and André Schaefer. Evaluating strategic support for information access in the DAFFODIL system. In Rachel Heery and Liz Lyon, editors, *Research and Advanced Technology for Digital Libraries. Proc. European Conference on Digital Libraries (ECDL 2004)*, Lecture Notes in Computer Science, Heidelberg et al., 2004. Springer.

[Klas *et al.*, 2006] Claus-Peter Klas, Hanne Albrechtsen, Norbert Fuhr, Preben Hansen, Sarantos Kapidakis, László Kovács, Sascha Kriewel, András Micsik, Christos Papatheodorou, Giannis Tsakonas, and Elin Jacob. A logging scheme for comparative digital library evaluation. In Julio Gonzalo, Costantino Thanos, M. Felisa Verdejo, and Rafael C. Carrasco, editors, *Research and Advanced Technology for Digital Libraries. Proc. of the 10th European Conference on Digital Libraries (ECDL 2006)*, Lecture Notes in Computer Science, pages 267–278, Heidelberg et al., September 2006. Springer.

[Klas *et al.*, 2008] Claus-Peter Klas, Sascha Kriewel, and Matthias Hemmje. An experimental system for adaptive services in information retrieval. In *Proceedings of the 2nd International Workshop on Adaptive Information Retrieval (AIR 2008)*, October 2008.

[Kriewel *et al.*, 2004] Sascha Kriewel, Claus-Peter Klas, André Schaefer, and Norbert Fuhr. Daffodil - strategic support for user-oriented access to heterogeneous digital libraries. *D-Lib Magazine*, 10(6), June 2004. http://www.dlib.org/dlib/june04/kriewel/06kriewel.html.

[Kwok, 1989] K. L. Kwok. A neural network for probabilistic information retrieval. In *Proceedings of the Twelfth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–30, New York, 1989. ACM.

[Malik *et al.*, 2006] Saadia Malik, Claus-Peter Klas, Norbert Fuhr, Birger Larsen, and Anastasios Tombros. Designing a user interface for interactive retrieval of structured documents — lessons learned from the inex interactive track. In *Proc. European Conference on Digital Libraries*, 2006.

[Peat and Willett, 1990] H. J. Peat and P. Willett. The limitations of term co-occurrence data for query expansion in document retrieval systems. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 42(5):378–383, 1990.

[Storkey and Valabregue, 1999] A. J. Storkey and R. Valabregue. The basins of attraction of a new hopfield learning rule. *Elsevier Science Ltd.*, 12:869–876, 1999.

[White, 2004] Ryen William White. *Implicit Feedback for Interactive Information Retrieval*. PhD thesis, University of Glasgow, 2004.

The Sense Folder Approach for Generic and Domain-Specific Retrieval Tasks

Ernesto William De Luca
Otto-von-Guericke University of Magdeburg
39106 Magdeburg
ernesto.deluca@ovgu.de

Frank Rügheimer
Biologie Systémique, Institut Pasteur
75015 Paris, France
frueghei@pasteur.fr

Abstract

In this work, we present and evaluate a new approach to semantic search. This approach is distinguished by pointing users to semantic concepts that offer possible refinements of their query. In parallel a combination of information retieval and machine learning strategies is applied to annotate and filter documents with respect to those semantic categories. In an outlook on recent work, we describe how the approach can be applied and extended as an aid to find information relevant to specific biological questions in publication databases.

1 Introduction

Search engines, such as Google and Yahoo have become an essential tool for the majority of Web users for finding information in the huge amount of documents contained in the Web. Even though, for most ad-hoc search tasks [Baeza-Yates and Ribeiro-Neto, 1999], they already provide a satisfying performance, certain fundamental properties still leave room for improvement. For example, users get lost in navigating the huge amount of documents available on the Web and are obliged to scan the list of all retrieved documents, in order to find the relevant ones. This can partially be attributed to the possibly misleading statistics that leads to semantically inhomogeneous result sets. Basically, results are computed from word frequencies and link structures, but other factors, such as sponsored links and ranking algorithms, are also taken into account.

More recent approaches try to categorize documents automatically with clustering methods. For instance, Vivísimo [Koshman *et al.*, 2006] organizes search results into categories (hierarchical clusters), basing on textual similarity. These methods only consider the word distribution in documents without taking into account linguistic criteria derived from the underlying query, such as different meanings of a term. Therefore, the assigned categories usually do not represent the categories a user is expecting for the query at hand.

In general, the search process starts when a user provides a list of keywords and the system returns a list of documents ordered by the degree of similarity to the applied query. This means that if the keywords are well chosen, an appropriate list of results is frequently provided. However, if the result list covers different meanings (if the search terms are ambiguous) or topics (if the search terms are used in different domains), then documents related to the corresponding categories appear rather unsorted in the result list.

Linguistic information (e.g. semantics) can provide valuable support for the user's search process. For instance, retrieved documents could be grouped by the meanings of the query. The user could choose one of these meanings and navigate only the documents related to it.

In addition, users search the Web and formulate their queries in their own language. But when they are unsuccessful, i.e. when their query does not match any results, they may also search and read documents in a foreign language [Peters and Sheridan, 2000].

In the light of the current lack of readily available tools that actively support researchers in finding desired information, given e.g. a gene name, we argue that semantic support would be a valuable addition to biological research tools. We therefore transfer the Sense Folder approach to biological data and set out to develop a search engine for biological publication databases.

1.1 Preliminaries

In order to better understand the semantic-based *Sense Folder approach* (see Section 2) presented in this paper, we first introduce some preliminary definitions (see Section 1.1) and related work (see Section 1.2). Then, the system architecture (see Section 2.1), semantic support methods (see Section 2.2) and the related user interface (see Section 2.5) are discussed. *Document classification* (see Section 2.3) and *clustering* (see Section 2.4) techniques used for filtering documents semantically are explained. Finally, the approach is compared and evaluated with respect to various baselines (see Section 3). The paper finishes with the discussion of ongoing research, where this semantic-based approach is applied for biological domain-specific tasks (see Section 4) and some concluding remarks (see Section 5).

Lexical resources, containing the different meanings of the words and the related linguistic relations, provide the semantic information needed for categorizing documents. For successful information retrieval it is crucial to represent documents in an adequate form with suitable attributes. In order to semantically compare documents, similarity measures have to be applied. The approach should be evaluated using appropriate performance measures.

Lexical Resources

Lexical resources are a special type of language resources [Cole *et al.*, 1997] that provide linguistic information about words. Lexical resources are used in this work for supporting the user with semantic information during the search process, as discussed in the following.

WordNet [Miller *et al.*, 1990; Fellbaum, 1998] is one of the most important English lexical resources available and can be used for text analysis and many related areas [Morato *et al.*, 2004], like word sense identification, disambiguation, and information retrieval [Vintar *et al.*, 2003]. WordNet provides a list of word senses for each word, organized into synonym sets (SynSets), each carrying exactly one meaning. Different relations link the SynSets to two types of linguistic relations, the first type is represented by lexical relations (e.g. synonomy, antonomy and polysemy), and the second by semantic relations (e.g. hyponomy and meronomy). Glosses (human descriptions) are often (about 70% of the time) associated with a SynSet [Ciravegna *et al.*, 1994].

We decided to use Wordnet for retrieving the meanings related to the queries and the related linguistic relations.

Vector Space Model

The vector space model [Salton, 1971; Salton and Lesk, 1971] is the most frequently used statistical model for adhoc retrieval and represents a user query q and a document d_i as vectors in a multi-dimensional linear space. Each dimension corresponds to characteristics of a word in a document (word occurrence, word frequency, etc.). For instance, if word occurrence is used, each dimension takes boolean values, while the use of (weighted) relative term or word frequency, such as tf or $tf \times idf$, leads to a real-valued vector space $\mathcal{V} = [0, 1]^n$.

Thus, the document vectors can be represented with attributes of terms in a document, such as term frequency (tf) or inverse document frequency (idf) [Salton and Buckley, 1988].

Cosine Similarity

Similarity between documents is assumed to coincide with the similarity in the vector space measured for instance using *cosine similarity* [Manning and Schütze, 1999]. This is a measure of similarity between two vectors of n dimensions computed by finding the angle between them. The approach relies on the assumption that a relevant document d_i and the corresponding query q are linked by common terms. These common word occurrences are expected to be reflected by a document vector $\vec{d_i}$ that is close to the query vector $\vec{q}$. That way, the task of finding relevant documents, given a query, is reduced to the identification of the document vectors that form the smallest angles with q. Therefore, the cosine similarity is defined as follows:

$$sim(d_i, q) = \frac{d_i \cdot q}{|\vec{d_i}| \times |\vec{q}|}. \qquad (1)$$

Performance Measures

Generally, queries are usually less than perfect for two reasons: first of all, they retrieve some irrelevant documents and secondly, they do not retrieve all the relevant documents. In order to evaluate the effectiveness of a retrieval system, different measures can be used [Baeza-Yates and Ribeiro-Neto, 1999]. The measure chosen for the evaluation of the classification performance in this work is the $accuracy$ that is the proportion of the total number of correct predictions.

1.2 Related Work

In this section related work that evaluate and compare different parameter settings is presented.

Agirre and Rigau [Agirre and Rigau, 1996] analyze WordNet relations for WSD and evaluate different combinations for disambiguating words using a conceptual density algorithm. They show that some relations such as meronymy (has-part relation) do not improve the performance as expected. They also point out that in WordNet not all semantic relations are available for all words, which might result in significant classification problems, since one disambiguating class might be described more specific than another class.

Larsen and Aone [Larsen and Aone, 1999] propose the application of clustering methods with a vector space model and tf or $tf \times idf$ document representations to overcome the information overload problem. They conclude that weighting terms by $tf \times idf$ works better than weighting by tf, except for corpora with a very small number of documents. But the influence of linguistic relations of words for classification is not taken into account.

In recent work Patwardhan and Pedersen [Patwardhan and Pedersen, 2006] employ so-called *context vectors* for Information Retrieval, including only WordNet glosses, because these descriptions are considered to contain content rich terms that better allow to distinguish concepts than a generic corpus.

2 Semantic-based Search and Disambiguation: The Sense Folder Approach

The idea of this approach is to use lexical resources in order to disambiguate/filter documents retrieved from the Web, given the different meanings (e.g. retrieved from lexical resources) of a search term, and the languages the users are able to speak. For achieving this goal different approaches are combined. *Word Sense Disambiguation* approaches are used for recognizing the contexts of the words contained in the query. *Document Categorization* techniques are used for collecting similar documents in semantic groups. *Semantic* and *Multilingual Text Retrieval* methods are developed because of the need of supporting humans to filter and retrieve relevant documents from the huge amount of data available using semantics and multilingual knowledge [De Luca and Nürnberger, 2006c; 2006b]. This section summarizes the core of this paper, the *Sense Folder Approach*. After defining a *Sense Folder*, the system architecture and the related user interface are described. Then, the classification and clustering methods used are discussed in more detail.

Sense Folder Definition Given a query term q, a *Sense Folder* is a container (prototype vector) that includes all selected linguistic information (linguistic context) of one sense of the query term retrieved from lexical resources.

2.1 System Architecture

Figure 1 gives an overview of the Sense Folder system architecture (and the related disambiguation process). The process starts after the user submits a query through the user interface (see [De Luca and Nürnberger, 2006a] and

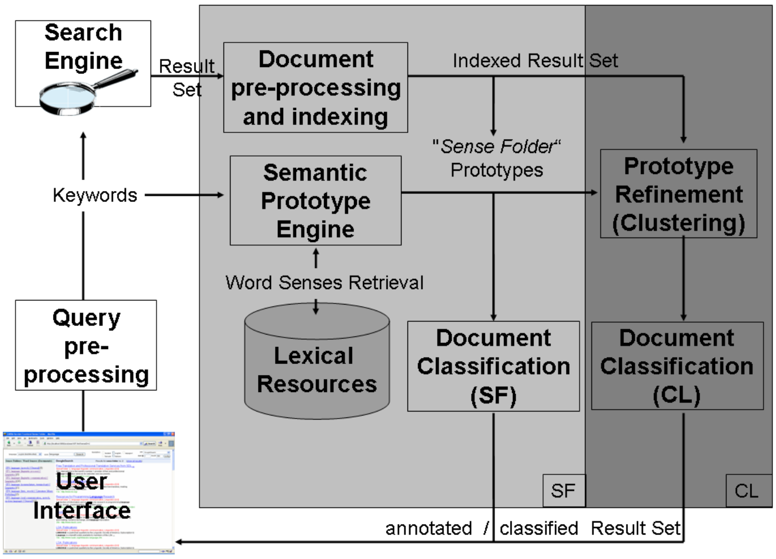

Search Engine
Result Set
Document pre-processing and indexing
Indexed Result Set
"Sense Folder" Prototypes
Keywords
Semantic Prototype Engine
Prototype Refinement (Clustering)
Word Senses Retrieval
Query pre-processing
Lexical Resources
Document Classification (SF)
Document Classification (CL)
SF
CL
User Interface
annotated / classified Result Set

Figure 1: Overview of the Sense Folder Approach.

Section 2.5). For every word contained in the query a pre-processing step is applied (see Section 2.2).

After the query has been processed, the user keywords are simultaneously sent to the search engine and to the *Semantic Prototype Engine*. While documents are retrieved, pre-processed and indexed, for every search term the different meanings of a term and the related linguistic relations are retrieved from the lexical resource. Using these linguistic relations a query term can be expanded with words defining the context for each of its meanings, thus forming *Sense Folders*. Based on this information, semantic prototype vectors describing each semantic class are constructed.

Then, based on this information for each document (retrieved from the search engine) the similarity to the Sense Folder prototypes is computed and the semantic class with the highest similarity to the considered document is assigned. This first categorization method is called *"pure" Sense Folder* (SF) classification approach (see Section 2.3).

Afterwards, clustering algorithms (see Section 2.4) are applied in order to fine tune the initial prototype vectors of each Sense Folder using the distribution of documents around the initial prototype vectors, i.e., we expect that in a Web search usually a subset of documents for each possible meaning of a search term is retrieved. Thus, each subset forms a cluster in document space describing one semantic meaning of this term. This additional clustering step (CL) has been introduced in order to enhance the semantic-based classification (only based on lexical resources) by considering also similarities in-between documents.

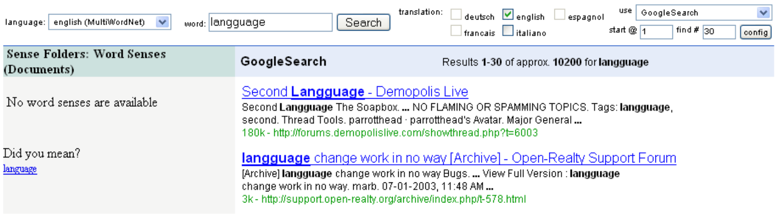
In contrast to the approach presented in [Agirre and Rigau, 1996] that only used a small window of words around the considered term in order to disambiguate its meaning, the assumption of this paper is that the meaning of a search term used in a Web page can be defined based on the whole document, since Web pages are usually very short and usually cover only one semantic topic. The assumption that words have only one sense per document in a given collocation is proven by experiments presented in [Gale *et al.*, 1992; Yarowsky, 1993].

2.2 Semantic-based Support: the Query Pre-Processing

In the following section three approaches that can be applied to queries and documents are described, in order to support users in the semantic-based searching process. Specifically, the goal is to improve the semantic search process; therefore several problems have to be addressed, before the semantic classification of documents is started. When users mistype in writing the query, the system has to be able to give correction alternatives, recognizing the etymology of the query words (e.g. using stemming methods) or recognizing named-entities to continue the semantic-based search. The semantic-based search differs from the "normal" search, because users are "redirected" to semantic concepts that could describe their query. This "redirection" is provided on the left side of the user interface (see Figure 2), where suggestions are generated by the system as described in the following.

Spelling Correction An important task for retrieving the relevant documents related to the query is to identify the misspelled words and correct them for a correct interpretation. In this work, the use of a spell-checker (implemented in a joint work [Ahmed *et al.*, 2007]) supports the user during the search process, not only because it performs an efficient correction, but also because it can "redirect" the user to a semantic search. Thus, if the user types a word that is not contained in the lexical resource used, the system can suggest other "similar" words (concepts), according to the words found by the spell checker. Then, a semantic classification is started using the words selected by the user [De Luca and Nürnberger, 2006c].

Stemming Because stemming methods are supposed to be suitable for reducing words to their base form, the Snowball stemmer [Porter, 2001] has been integrated. It includes a range of stemmers for different languages (e.g. the Porter stemmer for English, but also stemmers for French, German, Italian and Spanish). The aim of this approach is to improve performance merging similar variants of a word (sharing the same meaning) in one meaning. Stemming



language: english (MultiWordNet)
word: langguage
translation: □ deutsch ☑ english □ espagnol □ francais □ italiano
use GoogleSearch
start @ 1 find # 30

Sense Folders: Word Senses (Documents)

No word senses are available

Did you mean?
language

GoogleSearch Results **1-30** of approx. **10200** for **langguage**

Second **Langguage** - Demopolis Live
Second **Langguage** The Soapbox. **...** NO FLAMING OR SPAMMING TOPICS. Tags: **langguage**, second. Thread Tools. parrotthead · parrotthead's Avatar. Major General **...**
180k - http://forums.demopolislive.com/showthread.php?t=6003

langguage change work in no way [Archive] - Open-Realty Support Forum
[Archive] **langguage** change work in no way Bugs. **...** View Full Version : **langguage** change work in no way. marb. 07-01-2003, 11:48 AM **...**
3k - http://support.open-realty.org/archive/index.php/t-578.html

Figure 2: Semantic-based Support: the Query Pre-Processing.

methods do not analyze texts morphologically, but they try to reduce words in an etymological way to their base form; the stems are the result of such process. These stemmers can be used in order to help users in finding the base form of their keywords and "redirect" their search to the concept expressed in its base form. These base forms can be used by the users for the semantic-based search.

Named-Entity Recognition Because query words used for searching documents are not only common words, but represent also locations, organization, time expressions, and proper nouns, a named-entity recognizer (NER) has been added to the system, in order to support the user, if the search engine cannot disambiguate this kind of information. NER information is, in many cases, not ambiguous (e.g. a proper noun) and cannot be used for a semantic-based document search. The Stanford NER [Finkel *et al.*, 2005] can be used as a support for recognizing named-entities, directing the user to the semantic search. If the user types, for example, only the name "Java," the NER should recognize the meaning of this instance and suggest more than one disambiguation possibilities (e.g. when "Java." is related to the concept "island" or to the concept "programming language").

2.3 Sense Folder Classification (SF)

As discussed above, the user query is simultaneously submitted to the search engine, which is providing a set of search results and to the *Semantic Prototype Engine* that retrieves the linguistic information used for creating the vectors describing the disambiguating semantic classes. Each element in the vector corresponds to a term i in the document, while the size of the vector is defined by the number of words n occurring in the considered document collection (dictionary). The weights of the elements depend on the tf or $tf \times idf$ [Salton and Buckley, 1988] and can be combined with stemming methods. Once the vector space description for each document is computed, the documents are classified by computing the similarity to each prototype vector describing the disambiguating semantic classes and assigning the class with the highest similarity to the considered document. The cosine similarity measure (see Section 1.1) is used in this case for the "pure" (semantic) Sense Folder classification.

2.4 Sense Folder Clustering Methods (CL)

After the document vectors are assigned to their respective WordNet class, clustering methods are used to tune/refine the classification results. Clustering methods in this work use a small number of labeled documents (Sense Folders) with a large pool of unlabeled documents. Three clustering algorithms have been implemented and evaluated. The first is an unsupervised method (k-Means clustering [Manning and Schütze, 1999]), while the last two are semi-supervised (Expectation-Maximization Clustering [Dempster *et al.*, 1977] and Density-Based Clustering [Friedman and Meulman, 2004]).

#Word Sense (Synonyms) [Domain]
#0 chair (professorship) [Pedagogy]
#1 chair [Furniture]
#2 chair (electric chair, death chair, hot seat) [Law]
#3 chair (president, chairman, chairwoman, chairperson) [Person]

Table 1: WordNet noun collocation of the term "chair"

The k-Means clustering algorithm uses the number of classes obtained from WordNet for the number of clusters k as cluster centers. These classes are the so-called Sense Folders.

The Sense Folders are also used as "labeled data" for training the Expectation-Maximization clustering algorithm, i.e. in every cycle loop of the clustering process (in contrast to the k-Means clustering, where the Sense Folders are used only for the initialization) and the parameter λ is used as weight for the unlabeled/unclassified documents. If the parameter λ=0 the structure of unlabeled data is neglected and only the labeled data are considered for the estimation, while if the parameter λ=1 the information about the labeled class is neglected and only the structure is considered.

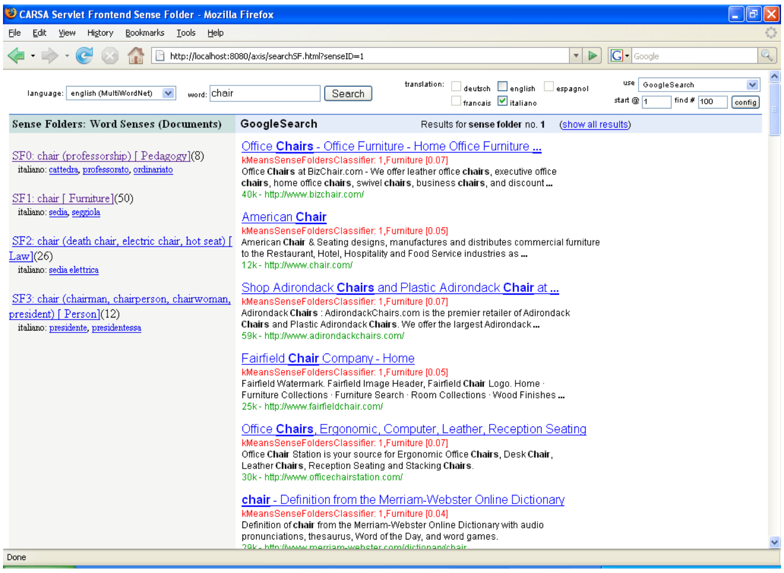
The initialization of the Density-based algorithm is the same of the k-Means algorithm. But the difference is due to the use of k neighbors and a parameter λ that is added. The use of such an algorithms is due to the assumption that data points that lie nearly together possess similar characteristics leads to an algorithm, where every data point is influenced by data points in its local neighborhood.

For a more detailed discussion about the clustering algorithms, the reader should refer to [De Luca, 2008].

2.5 Sense Folder User Interface

The semantic-based approach presented in this paper should simplify the search process by providing users with explicit information about ambiguities and this enables them to easily retrieve the subset of documents they are looking for. Labels defining the disambiguating classes are added to each document of the result set. This semantic information is assigned by the Sense Folder classification and clustering methods and appended as semantic annotation to the document as shown in Figure 3.

Thus, the visualization of such additional information gives the possibility to the user to filter the relevant query-related results by semantic class. For instance, if a user types, for example, the word "chair", he/she has the possibility to obtain four different semantic classes based on the noun collocations of this word (included in WordNet) as shown in Table 1. These classes represent the different meanings of the search terms given by the user and are pro-


CARSA Servlet Frontend Sense Folder - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/axis/searchSF.html?senseID=1
Google
Search
config
Done

language: english (MultiWordNet) word: chair

translation: □ deutsch □ english □ espagnol □ francais ☑ italiano

use GoogleSearch start @ 1 find # 100

Sense Folders: Word Senses (Documents)

SF0: chair (professorship) [Pedagogy](8)
italiano: cattedra, professorato, ordinariato

SF1: chair [Furniture](50)
italiano: sedia, seggiola

SF2: chair (death chair, electric chair, hot seat) [Law](26)
italiano: sedia elettrica

SF3: chair (chairman, chairperson, chairwoman, president) [Person](12)
italiano: presidente, presidentessa

GoogleSearch Results for **sense folder** no. **1** (show all results)

Office **Chairs** - Office Furniture - Home Office Furniture ...
kMeansSenseFoldersClassifier: 1,Furniture [0.07]
Office **Chairs** at BizChair.com - We offer leather office **chairs**, executive office **chairs**, home office **chairs**, swivel **chairs**, business **chairs**, and discount ...
40k - http://www.bizchair.com/

American **Chair**
kMeansSenseFoldersClassifier: 1,Furniture [0.05]
American **Chair** & Seating designs, manufactures and distributes commercial furniture to the Restaurant, Hotel, Hospitality and Food Service industries as ...
12k - http://www.chair.com/

Shop Adirondack **Chairs** and Plastic Adirondack **Chair** at ...
kMeansSenseFoldersClassifier: 1,Furniture [0.07]
Adirondack **Chairs** : AdirondackChairs.com is the premier retailer of Adirondack **Chairs** and Plastic Adirondack **Chairs**. We offer the largest Adirondack ...
59k - http://www.adirondackchairs.com/

Fairfield **Chair** Company - Home
kMeansSenseFoldersClassifier: 1,Furniture [0.05]
Fairfield Watermark. Fairfield Image Header, Fairfield **Chair** Logo. Home · Furniture Collections · Furniture Search · Room Collections · Wood Finishes ...
25k - http://www.fairfieldchair.com/

Office **Chairs**, Ergonomic, Computer, Leather, Reception Seating
kMeansSenseFoldersClassifier: 1,Furniture [0.07]
Office **Chair** Station is your source for Ergonomic Office **Chairs**, Desk **Chair**, Leather **Chairs**, Reception Seating and Stacking **Chairs**.
30k - http://www.officechairstation.com/

chair - Definition from the Merriam-Webster Online Dictionary
kMeansSenseFoldersClassifier: 1,Furniture [0.04]
Definition of **chair** from the Merriam-Webster Online Dictionary with audio pronunciations, thesaurus, Word of the Day, and word games.
29k - http://www.merriam-webster.com/dictionary/chair

Figure 3: Semantic-based Search Engine.

vided on the left side of the interface (see Figure 3). If a user selects one of these meanings (e.g. the Sense Folder meaning SF1), the documents related to this concept are shown/filtered. Thus, users do not need to scan all documents, but they can browse only the documents related to the SF1 meaning of their query.

3 Evaluation

This section summarizes the results of a detailed fine-grained evaluation of different parameter settings. Various linguistic relations are combined with the different document attributes explained in Section 1.1. These are measured and compared to each other in order to recognize their best combination. Sense Folder classification and clustering methods are added and evaluated against three different baselines (*Random*, *First Sense* and *Most Frequent Sense*).

3.1 Baselines

According to WSD evaluation tasks, the approach has been evaluated in a fine-grained framework that considers the distinction of word senses on a more detailed level (all senses are included). This evaluation has been combined within three baselines described in the following:

- A *Random Baseline* (`Random`) assuming a uniform distribution of the word senses. This baseline provides a simple boundary for classification performance.
- A *First Sense Baseline* (`FS`), i.e. the score achieved by always predicting the first word sense, according to a given ranking, of a given word in a document collection. The First Sense baseline is often used for supervised WSD systems [McCarthy *et al.*, 2004]. This baseline is based, for this work, on the first word sense of WordNet contained in the Sense Folder Corpus [De Luca, 2008].
- A *Most Frequent Sense Baseline* (`MFS`) based on the highest a-posteriori word sense frequency, given a word in a document collection, i.e. the score of a *theoretically* best result, when consistently predicting the same word sense for a given word. This baseline is based on the highest frequency of a word sense contained in a document collection. It is the best possible result score when consistently predicting one sense. It is often used as a baseline for evaluating WSD approaches and very difficult to outperform [McCarthy *et al.*, 2004].

3.2 Corpora Analysis

In order to evaluate this approach we analyzed different corpora to check if they were appropriate for the evaluation problem at hand. Many collections are already available in order to measure the effectiveness of information retrieval systems. Examples are given by the *Reuters* Corpus (RCV1), containing a large collection of high-quality news stories [Rose *et al.*, 2002], or the Reuters-21578 and Reuters-22173 data being the most widely test collection used for text categorization. Another collection is the *Text REtrieval Conference* (TREC) data collection, having the purpose to support information retrieval research by providing an infrastructure for large-scale evaluation of text retrieval methodologies. Because our purpose is not only evaluating an information retrieval system, but also a semantic information retrieval system, these data sets are unfortunately not appropriate for this task. They do not provide any semantic information based on a given query word, resulting that they are a document- and not query-oriented collection. No WordNet annotations are included and the "one sense per document" assumption is not fulfilled, because more topics can be covered in one document.

Since none of the available benchmark collections was appropriate for our task, we decided to use the multilingual *Sense Folder Corpus* created for evaluating semantic-based retrieval systems [De Luca, 2008]. This is a small bilingual (english and italian) hand-tagged corpus of 502 documents retrieved from Web searches created using Google queries. A single ambiguous word ("argument, bank , chair, network, rule" in englisch and "argomento, lingua , regola, rete, stampa" in italian) has been searched and related documents (approx. the first 60 documents for every keyword) have been retrieved. Every document contained in the collection has been annotated with only one WordNet domain

Linguistic Relations	Document Encoding		Clustering
Synonyms (`Syn`)			No Clustering (SF)
Domain (`Dom`)	Stemming (`Stem`)	tf (`Tf`)	K-Means Clustering (KM)
Hyponyms (`Hypo`)	No Stemming (`NoStem`)	$tf \times idf$ (`TfxIdf`)	Modified EM Clustering (EM)
Hyperonyms (`Hyper`)			Density-Based Clustering (DB)
Coordinate Terms (`Coord`)			
Domain Hierarchy (`DomH`)			
Human descriptions (`Gloss`)			

Table 2: Components of Fine-Grained Evaluation Procedures

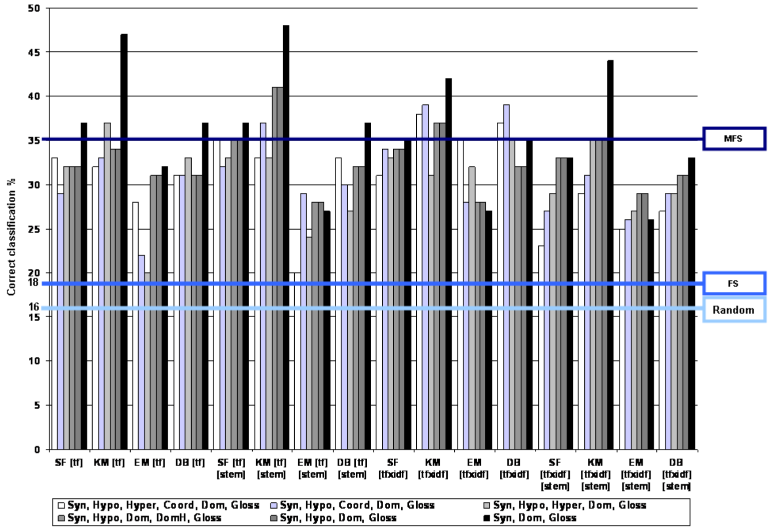


Figure 4: Sense Folder Fine-grained Accuracy Evaluation Results of different parameter settings with baselines.

label and one MultiWordNet query-dependent word sense label, respecting also the “one sense per document” assumption [Gale *et al.*, 1992; Yarowsky, 1993].

3.3 Parameter Settings

Table 2 outlines all components used for the fine-grained evaluation presented in this paper. Different combinations of linguistic relations (synonyms, coordinate terms, hyperonyms, hyponyms, glosses and semantic domains, and the semantic domain hierarchy) are taken into account, in order to assign documents to the semantic class they belong to.

Different encodings are considered in order to find an optimal representation. The tf and $tf \times idf$ encoding, as well as the *stemming* vs. *not stemming* term features, describe different vector spaces for the document classification. The resulting parameter settings are: tf-based (`Tf`) and respective stemmed one (`Tf+Stem`), $tf \times idf$-based (`TfxIdf`) and respective stemmed one (`TfxIdf+Stem`).

Three Sense Folder clustering methods have been implemented. The k-Means clustering algorithm (`KM`) does not require any parameters, because the number k of cluster centers, corresponds to the number of word senses available in WordNet used as initial prototypes for the clustering process.

The DB Clustering method (`DB`) uses Sense Folders as initial prototypes. The parameter λ has been set to 0.9 and the n parameter that represents the number of neighbors to be considered, is set to 2.

The Expectation-Maximization(`EM`)-λ algorithm adopts the Sense Folders in the role of “labeled data,” whereas the vectors representing the documents supply the “unlabeled data”. The weight parameter λ for the unclassified data points is set to 0.9.

The parameter settings for the last two clustering algorithms (EM-λ and DB) have been set according to the evaluation results presented in [Honza, 2005].

3.4 Fine-grained Accuracy Evaluation

Analyzing the results presented in Figure 4, we can notice that a slight overall improvement is shown when stemming methods are applied in conjunction with the tf measure. We can see that the “pure” Sense Folder classification in some cases is already sufficient for classifying documents in the correct meaning. But in most cases clustering methods improve classification considerably.

When the automatic classification is compared within the baselines, we can see that all combinations outperform all “Random” and “First Sense” baselines.

Analyzing the linguistic relations in more detail, we can notice that the use of hyperonyms or hyponyms negatively influence the classification performance. Normally, a hyperonym should be the broader term of a given word that generalize the related linguistic context. But these terms included in WordNet are at the end too general and make the disambiguation of word senses more difficult. As a rule, a hyponym should narrow down the distinct word senses describing the search word more specifically; but these WordNet terms are not significant enough to split them.

When such linguistic information is combined with clustering methods, in some cases, the classification performance is strongly enhanced, because similar documents are recognized. Sometimes this semantic information already contained in lexical resources is sufficient to recognize the linguistic context of a document given a query, so that clus-

Notion	General IR	Biological Domain
ambiguous description	word	gene name
referenced entity	word sense	gene/protein/protein acting in particular biological role
relations between entities	WordNet	Gene Ontology

Table 3: Corresponding Notions for General Information Retrieval and Gene Names in Biological Domain

tering methods are not needed or their use negatively affects the classification.

The recognition of the correct word sense given a query for retrieving only relevant documents is fundamental. This is needed to better support users in the search process, showing only the filtered relevant documents.

Summarizing, the fine-grained classification works better with the tf-based document representation and stemming methods than with the $tf \times idf$-based document representation. This is most likely to be attributed to the idf measure that cannot be estimated very good and be meaningful adopted, because the document collection is still relative small.

4 Semantic-based Support in Biology: The use of the GENE Ontology

Many of the difficulties encountered in web information retrieval are paralleled when querying biological databases. With the recent development in experimental techniques, finding the most relevant pieces of information in the growing pool of published biological knowledge becomes increasingly difficult. From this situation arises the need for suitable search engines that support biologists in submitting queries that are adapted to very particular information needs.

The products of many genes, for instance, are used in two or more often otherwise unrelated roles in the organism. Such multiple roles arise, for instance, from alternative splicing (single gene gives rise to several proteins) or due to the coded proteins possessing several active sites. In biological databases this is reflected by pairs of biological process annotations for the same element with neither term being registered as a specialization of the other in the Gene Ontology [Ashburner *et al.*, 2000]. A researcher, however, would often only be interested in material on one of these roles, so the possibility to additionally restrict queries is desirable.

A second problem arises from the fact that many genes were discovered and named independently before being identified as referring to identical or analogue entities. This leads to synonymous gene names. In practice such synonyms are treated by mapping known alternative gene names to a standard descriptor. In rare cases two or more different genes share at least one name though (gene homonyms). This situation results in an ambiguity comparable to the different biological functions discussed above, though this time the ambiguity also extends to the genetic level as well, as the genes in questions correspond to different locations in the genome.

In all those cases the Sense Folder approach allows to guide and refine searches allowing to focus on results from the desired context only (compare Section 2.5). Moreover, with the Gene Ontology, a structure that allows to expand the list of search terms is already available.

Finally we need to address the problem of associating potential (query results) with individual Sense Folders. Fortunately this task is completely analogous to the general case so the approach described in Section 2.3 can be applied. The observed analogies are summarized in Table 3.

5 Concluding Remarks

In this paper, an approach for semantic-based search and support has been presented. It combines different techniques to provide the extended functionality of "directing" users to semantic concepts that help increase the specificity of their queries. On the application level, semantic support is provided via additional components that have been integrated into the user interface of search engines. We evaluated the performance of the presented semantic-based approach, and conclude that the accurate selection of linguistic relations, clustering methods and document encodings strongly influences the fine-grained classification results. All baselines are outperformed and best results are achieved when combining the linguistic relations (synonyms, domains and glosses) with the k-Means Clustering algorithm, representing the document vectors as tf-based vectors and applying stemming methods. An outlook about the employment of such an approach for biological tasks is discussed and the tasks in this domain-specific context found to be comparable to those of the more general information retrieval problem.

References

[Agirre and Rigau, 1996] Eneko Agirre and German Rigau. Word sense disambiguation using conceptual density. In *Proceedings of COLING'96*, pages 16–22, 1996.

[Ahmed *et al.*, 2007] Farag Ahmed, Ernesto William De Luca, and Andreas Nürnberger. MultiSpell: an N-Gram Based Language-Independent Spell Checker. In *Proceedings of Eighth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2007)*, Mexico City, Mexico, 2007.

[Ashburner *et al.*, 2000] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nat Genet*, 25(1):25–29, May 2000.

[Baeza-Yates and Ribeiro-Neto, 1999] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, Addison-Wesley, New York, 1999.

[Ciravegna *et al.*, 1994] Fabio Ciravegna, Bernardo Magnini, Emanuele Pianta, and Carlo Strapparava. A project for the construction of an italian lexical knowledge base in the framework of wordnet. Technical Report IRST 9406-15, IRST-ITC, 1994.

[Cole *et al.*, 1997] R. Cole, J. Mariani, H. Uszkoreit, A. Zaenen, and V. Zue. *Survey of the State of the Art in*

Human Language Technology. Center for Spoken Language Understanding CSLU, Carnegie Mellon University, Pittsburgh, PA, 1997.

[De Luca and Nürnberger, 2006a] Ernesto William De Luca and Andreas Nürnberger. A Word Sense-Oriented User Interface for Interactive Multilingual Text Retrieval. In *Proceedings of the Workshop Information Retrieval In conjunction with the LWA 2006, GI joint workshop event 'Learning, Knowledge and Adaptivity'*, Hildesheim, Germany, 2006.

[De Luca and Nürnberger, 2006b] Ernesto William De Luca and Andreas Nürnberger. Rebuilding Lexical Resources for Information Retrieval using Sense Folder Detection and Merging Methods. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, Genova, Italy, 2006.

[De Luca and Nürnberger, 2006c] Ernesto William De Luca and Andreas Nürnberger. The Use of Lexical Resources for Sense Folder Disambiguation. In *Workshop Lexical Semantic Resources (DGfS-06)*, Bielefeld, Germany, 2006.

[De Luca, 2008] Ernesto William De Luca. *Semantic Support in Multilingual Text Retrieval*. Shaker Verlag, Aachen, Germany, 2008.

[Dempster *et al.*, 1977] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, Series B(39(1)):1–38, 1977.

[Fellbaum, 1998] Christiane Fellbaum. *WordNet, an electronic lexical database*. MIT Press, 1998.

[Finkel *et al.*, 2005] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[Friedman and Meulman, 2004] Jerome H. Friedman and Jacqueline J. Meulman. Clustering objects on subsets of attributes (with discussion). *Journal Of The Royal Statistical Society Series B*, 66(4):815–849, 2004.

[Gale *et al.*, 1992] William A. Gale, Kenneth W. Church, and David Yarowsky. One sense per discourse. In *Proceedings of the 4th DARPA Speech and Natrual Language Workshop*, pages 233–237, 1992.

[Honza, 2005] Frank Honza. Clustering mit a-priori annahmen über clusterspezifische attributwichtigkeiten. Master's thesis, University of Magdeburg, Germany, 2005.

[Koshman *et al.*, 2006] Sherry Koshman, Amanda Spink, and Bernard J. Jansen. Web searching on the vivisimo search engine. *J. Am. Soc. Inf. Sci. Technol.*, 57(14):1875–1887, 2006.

[Larsen and Aone, 1999] Bjornar Larsen and Chinatsu Aone. Fast and effective text mining using linear-time document clustering. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 16–22, New York, NY, USA, 1999. ACM Press.

[Manning and Schütze, 1999] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Boston, USA, 1999.

[McCarthy *et al.*, 2004] D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. Finding predominant word senses in untagged text. In *42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain, 2004.

[Miller *et al.*, 1990] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Five papers on WordNet. *International Journal of Lexicology*, 3(4), 1990.

[Morato *et al.*, 2004] Jorge Morato, Miguel Angel Marzal, Juan Llorens, and José Moreiro. Wordnet applications. In Masaryk University, editor, *Proceedings of the 2nd Global Wordnet Conference 2004*, 2004.

[Patwardhan and Pedersen, 2006] Siddharth Patwardhan and Ted Pedersen. Using WordNet Based Context Vectors to Estimate the Semantic Relatedness of Concepts. In *Proceedings of the EACL 2006 Workshop Making Sense of Sense - Bringing Computational Linguistics and Psycholinguistics Together*, pages 1–8, Trento, Italy, 2006.

[Peters and Sheridan, 2000] Carol Peters and Páraic Sheridan. Multilingual information access. In *Lectures on Information Retrieval, Third European Summer-School, ESSIR 2000, Varenna, Italy*, 2000.

[Porter, 2001] M.F. Porter. Snowball: A language for stemming algorithms. Technical report, Open Source Initiative OSI, 2001.

[Rose *et al.*, 2002] T.G. Rose, M. Stevenson, and M. Whitehead. The reuters corpus volume 1 - from yesterday's news to tomorrow's language resources. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, Las Palmas de Gran Canaria, 2002.

[Salton and Buckley, 1988] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[Salton and Lesk, 1971] G. Salton and M. E. Lesk. *Computer evaluation of indexing and text processing*, page 143180. Prentice-Hall, Inc. Englewood Cliffs, 1971.

[Salton, 1971] G. Salton. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

[Vintar *et al.*, 2003] S. Vintar, P. Buitelaar, and M. Volk. Semantic relations in concept-based cross-language medical information retrieval. In *Proceedings of the ECML/PKDD Workshop on Adaptive Text Extraction and Mining*, Croatia, 2003.

[Yarowsky, 1993] David Yarowsky. One sense per collocation. In *Proceedings of ARPA Human Language Technology Workshop*, 1993.

A comparison of sub-word indexing methods for information retrieval

Johannes Leveling
Centre for Next Generation Localisation (CNGL)
Dublin City University, Dublin 9, Ireland
johannes.leveling@computing.dcu.ie

Abstract

This paper compares different methods of sub-word indexing and their performance on the English and German domain-specific document collection of the Cross-language Evaluation Forum (CLEF). Four major methods to index sub-words are investigated and compared to indexing stems: 1) sequences of vowels and consonants, 2) a dictionary-based approach for decompounding, 3) overlapping character n-grams, and 4) Knuth's algorithm for hyphenation.

The performance and effects of sub-word extraction on search time and index size and time are reported for English and German retrieval experiments. The main results are: For English, indexing sub-words does not outperform the baseline using standard retrieval on stemmed word forms (–8% mean average precision (MAP), –11% geometric MAP (GMAP), +1% relevant and retrieved documents (rel_ret) for the best experiment). For German, with the exception of n-grams, all methods for indexing sub-words achieve a higher performance than the stemming baseline. The best performing sub-word indexing methods are to use consonant-vowel-consonant sequences and index them together with word stems (+17% MAP, +37% GMAP, +14% rel_ret compared to the baseline), or to index syllable-like sub-words obtained from the hyphenation algorithm together with stems (+9% MAP, +23% GMAP, +11% rel_ret).

1 Introduction

Splitting up words into sub-words is a technique which is frequently used to improve information retrieval (IR) performance. The main idea behind sub-word indexing is to break up long words into smaller indexing units. These indexing units can be found by methods such as decompounding words into lexical constituent words or splitting words into character n-grams of a fixed size. In some languages like German, compounds are written as a single word. Thus, if a German query or document contains a compound word like *"Kinderernährung"* (nutrition of children), the words *"Kind"* (child) and *"Ernährung"* (nutrition) will not match and result in low recall. Splitting the compound word and finding smaller indexing units will make a match more likely and yield a higher recall. For instance, a decompounding process may identify the constituent words *"Kinder"* (children) and *"Ernährung"*, which can be used in a query to achieve a higher IR performance. Linguistically oriented approaches aim at breaking up compound words into constituent words. Other approaches to generate sub-words do not build on the notion that sub-words must be valid words of the language (e.g. character n-grams).

For languages with a rich morphology (like Finnish, Dutch or German), a linguistically motivated decomposition of words has been widely recognised as a method to improve IR performance [Braschler and Ripplinger, 2003; Chen and Gey, 2004]. In languages such as English, compounds are typically written as separate words and their constituents can be easily identified.

However, creating resources such as dictionaries is expensive and time-consuming and dictionaries depend on language and domain. The most extreme knowledge-light approach at decompounding, overlapping character n-grams, has extreme requirements for index space due to combining grams for different values of n [McNamee, 2001; McNamee and Mayfield, 2007]. Decompounding methods should in the best case be efficient and effective, i.e. they should be inexpensive (i.e. not rely on external resources), largely independent of a particular domain; and adaptable to many languages. One aim of this paper is to help identify such an approach for decompounding words.

The contribution of this paper is the quantitative evaluation of four different sub-word indexing methods. The performance of the methods and their combination with stemming is compared for a compounding and a non-compounding language i.e., German and English. Sub-word indexing based on consonant-vowel-consonant sequences has primarily been used in speech retrieval and not in domain-specific information retrieval. Two of the variants of this approach (consonant-vowel sequences and vowel-consonant sequences) are novel. Knuth's algorithm for hyphenation has not been applied before to identify syllable-like sub-words as indexing units. Effects of sub-word indexing on the index size and on indexing time and search time are rarely discussed.

The rest of this paper is organised as follows: Section 2 introduces the sub-word identification techniques used in the IR experiments in this paper. Section 3 gives an overview over related work where approaches to decompounding have been employed. Section 4 describes the experimental setup for the experiments. Section 5 discusses the influence of sub-word indexing on retrieval performance, search time, and indexing time and space and provides a topic analysis. Section 6 concludes with a description of future work.

2 Identifying sub-words

The information retrieval experiments describes in this paper are conducted on German and English queries and documents to investigate the performance of sub-word identification for a compound-rich and a non-compounding language. Four different approaches to sub-word indexing are evaluated and compared to the baseline of indexing stems (stem):[1]

1. consonant-vowel sequences (CV) and derived methods, including vowel-consonant sequences (VC), consonant-vowel-consonant sequences (CVC), and vowel-consonant-vowel sequences (VCV);
2. a dictionary-based approach to identify constituent words of compound words (DICT);
3. syllable-like character sequences determined by Knuth's algorithm for hyphenation (HYPH); and
4. overlapping character n-grams (3-grams, 4-grams, and 5-grams).

Table 1 shows results of applying sub-word identification to the German word *"Informationssuche"* (information retrieval). The following subsections provide a more detailed description of these sub-word indexing techniques.

2.1 Dictionary-based decompounding

Dictionary-based decomposition of a word typically involves repeatedly determining whether prefix strings of a compound are valid words by looking them up in a dictionary. Many decompounding approaches used for German IR consider only the most frequent rule or rules of word formation. For example, the word *"Betriebskosten"* (operating costs) consists of two constituents, *"Betrieb"* and *"Kosten"*, connected by a so called Fugen-*s*. This connection represents one of the most frequent patterns in German compound word formation.

Dictionary-based decompounding is quite robust to some linguistic effects in the German language. For example, some compounds contain constituents in their plural form (e.g. *"Gänsefleisch"* (literally: geese meat)), which will be normalised to the same base as the words in singular form after stemming is applied (e.g. *"Gans"* (goose) and *"Fleisch"* (meat)). Some compounds should not be split into their constituents at all (e.g. *"Eisenbahn"* (railway) oder *"Lieblingsgetränk"* (favourite drink)), but these cases are rare and can be treated by using exception lists. Decompounding even allows for ambiguous results for the same compound. For example, *"Arbeitsamt"* (employment bureau), can be split into *"Arbeit"* (work), Fugen-*s*, and *"Amt"* (bureau) or into *"Arbeit"* and *"Samt"* (velvet). Ambiguities are typically resolved by a left-to-right, longest match preference.

However, dictionary-based decompounding requires language-specific dictionaries and additional processing time for successively looking up potential constituents in the dictionary to determine if they form valid words.

2.2 Consonant-vowel sequences

The Porter stemming algorithm [Porter, 1980] is a rule-based heuristic to normalise words to index terms by suffix removal. As a by-product, it computes the M-measure, a count roughly corresponding to the number of syllables in the word.[2] The M-measure is defined via the number of consonant-vowel-consonant sequences (short: CVC sequences) in a word. The set of vowels differs from language to language: In German, vowels are *"a"*, *"e"*, *"i"*, *"o"*, *"u"* (not counting letters with diacritical marks); in English, vowels also include *"y"* if preceded by a consonant. Other languages such as Arabic or Hebrew have no letters to represent vowels. The computation of the M-measure in the Porter stemmer can be easily adapted to generate sub-words, i.e. by adding a sub-word to a list each time M is increased. The M-measure can also be calculated for words in other languages by defining the corresponding set of vowels. The Snowball string processing language[3] provides stemmers for a range of different languages.

A CVC sequence is the longest match of a sequence of zero or more consonants (C), followed by zero or more vowels (V), followed by one or more consonants in a word. Three variants of these character sequences can be defined accordingly (VCV, CV, and VC sequences) and are investigated in this paper, too.

From an IR perspective, CVC sequences offer a cheap alternative to a complex morphologic analysis of words. As stemming has become a standard approach to normalise indexing terms, the modification of a stemmer to produce CVC sequences would require little additional cost.

2.3 Overlapping character n-grams

Words can be broken up into sequences of characters of a fixed size n to form character n-grams. If $n-$grams are allowed to start at every character position (instead of one n-gram for every n characters), the n-grams will partially overlap. Some variants of this method include adding an extra character as a special word boundary marker to n-grams from the beginning and end of a word. Following this approach and the character "|" as a boundary marker, the set of 4-grams for the noun *"Lichter"* includes the gram *"|lich"* from the beginning of the word and allows to distinguish it from the common adjectival ending *"lich|"*.

In another approach, the full text is regarded as a single string and not broken down into words before calculating n-grams. Whitespace characters are not discarded and become part of the character n-grams, which can span word boundaries.

2.4 Knuth's hyphenation algorithm

Knuth's hyphenation algorithm was developed by Knuth and Liang for dividing words at line breaks for the TeX/LaTeX typesetting tool [Liang, 1983; Knuth, 1984]. It is well documented and has been used in the document formatting system groff, in the PostScript language, and in the programming language Perl. At its core are sets of language-specific patterns. The patterns are employed to identify positions at which a line break can occur and a word can be divided. In this paper line break positions between two characters are interpreted as positions marking sub-word boundaries for sub-word identification,

3 Related Work

Decompounding is a successful method to improve retrieval performance in IR. There have been numerous re-

[1] Stemming can be viewed as a way to identify a single sub-word within a word by affix removal and is considered as a baseline for sub-word indexing.

[2] In a pre-test, the number of syllables was calculated correctly in about 93% using the M-measure on a test set of about 30,000 manually annotated words. Most errors resulted from foreign expressions and proper nouns.

[3] http://snowball.tartarus.org/

Table 1: Examples for splitting the German word *"Informationssuche"* into sub-words with different methods.

method	sub-words	# sub-words
stem	informationssuch	1
CV	i, nfo, rma, tio, nssu, che	6
VC	inf, orm, at, ionss, uch, e	6
CVC	inf, nform, rmat, tionss, nssuch	5
VCV	info, orma, atio, onssu, uche	5
DICT	information, suche	2
HYPH	in, for, ma, ti, ons, su, che	7
3-grams	inf, nfo, for, orm, rma, mat, ati, tio, ion, ons, nss, ssu, suc, uch, che	15
4-grams	info, nfor, form, orma, rmat, mati, atio, tion, ions, onss, nssu, ssuc, such, uche	14
5-grams	infor, nform, forma, ormat, rmati, matio, ation, tions, ionss, onssu, nssuc, ssuch, suche	13

trieval experiments using simple rule-based or dictionary based approaches to decompounding German words. Note: Most researchers report performance gain comparing sub-words originating from stems to a baseline with indexing unprocessed word forms. This results in better performance values (as effects of stemming are included in sub-words experiments), but make a comparison with other retrieval experiments more difficult.

Kamps et al. perform information retrieval experiments including decompounding to documents from the CLEF 2003 collection in nine languages. They report a 7.5% increase in MAP for an experiment on the German document collection including dictionary-based decompounding over baseline with stems and a 13.0% increase for 4-grams [Kamps *et al.*, 2003]. Results for decompounding English documents are not given.

Chen and Gey use dictionary-based decompounding to the CLEF 2001 and 2002 test collections [Chen and Gey, 2004]. Decompounding is based on computing the probability of the best splitting sequence based on the frequency of constituents [Chen, 2003]. For monolingual German retrieval experiments, they report a 12.7% increase in MAP and 4.6% in relevant retrieved documents for the 2001 data (13.8% and 13.1% for 2002 data, respectively) when indexing stemmed compounds together with their constituents compared to an experiment using only stems.

Daumke et al. apply MorphoSaurus as a text processing tool to documents [Daumke, 2007; Daumke *et al.*, 2007]. MorphoSaurus breaks down words into sub-words based on a dictionary with pseudo-morphological word elements. The sub-word segmentation of a word is determined automatically based on a manually created list of sub-words. For the English OSHUMED test collection, they achieve 5% increase in MAP compared to a stemming baseline; for German GIRT data, a decrease of 19.5% in MAP, and for German data from the image retrieval task ImageCLEF, an increase from 0.0343 to 0.0403 MAP (+17.5%).

Glavitsch and Schäuble extract CVC sequences as indexing features for retrieval of speech documents [Glavitsch and Schäuble, 1992; Schäuble and Glavitsch, 1994]. They select features based on document and collection frequency, and discrimination value. This indexing method performs slightly better than one using stopword removal and stemming. Similarly, Ng performs experiments on spoken documents for English, achieving 28% performance increase when combining sub-words indexing with error compensation routines [Ng, 2000]. CVC sequences are often used as indexing units for speech retrieval, even for non-European languages.

Braschler and Ripplinger give an overview about stemming and decompounding for German [Braschler and Ripplinger, 2003]. They perform IR experiments on data from CLEF for the ad-hoc retrieval track. They apply a variety of approaches for stemming and decompounding – including commercial solutions – and achieve a performance gain of up to 60.4% MAP and 30.3% for the number of relevant retrieved documents in comparison to indexing raw word forms (not stems).

McNamee performs retrieval experiments using overlapping character n-grams as indexing units [McNamee, 2001]. He reports performance results for indexing a combination of 2-grams, 3-grams, and 4-grams for English, Chinese, Japanese, and Korean. Results show that n-grams can achieve similar or superior performance in comparison to standard indexing techniques, even for non-compounding languages and for cross-lingual retrieval [McNamee and Mayfield, 2007].

To the best of the authors' knowledge, hyphenation algorithms or syllabification have not been applied to find sub-words for information retrieval on written documents before.

4 Experimental Setup and System Description

The retrieval experiments in this paper are based on data from the German Indexing and Retrieval Test database (GIRT) [Kluck, 2005] used in the domain-specific track at CLEF (Cross Language Retrieval Forum). The document collections in German and English consist of 151,319 documents from the GIRT4 database.[4] The topics include the 150 German and English topics from the domain-specific track at CLEF from 2003 to 2008 (25 topics each year), together with official relevance assessments.

A GIRT document contains metadata on publications from the social sciences, represented as a structured XML document. The metadata scheme defines 14 fields, including abstract, authors, classification terms, controlled terms, date of publication, and title. Figure 1 shows an excerpt from a sample document.

A GIRT topic resembles topics from other retrieval campaigns such as TREC. It contains a brief summary of the information need (topic title), a longer description (topic description), and a part with information on how documents are to be assessed for relevance (topic narrative). Retrieval

[4] In 2006, 20,000 abstracts from Cambridge Scientific Abstracts were added to the English GIRT document collection. As there are no relevance assessments available for topics from before 2006, these documents were discarded for the experiments.

queries are typically generated from the title (T) and description (D) fields of topics. Figure 2 shows a sample topic.

For each GIRT topic, relevant documents have been assessed by pooling submissions from systems participating in the domain-specific track at CLEF, resulting in a total of more than 80,000 relevance assessments for German documents (68,000 for English documents, respectively), including 16,200 German relevant documents for 150 topics (14,162 for English). The experimental results in this paper are based on the complete set of German and English topics and their corresponding relevance assessments.

The experiments were conducted with the following system setup. Lucene[5] was employed to preprocess the topics and documents and to index and search the document collection. The document structure was flattened into a single index by collecting the abstract, title, controlled terms and classification text as content and discarding the rest (e.g. author, publication-year, and language-code). The following preprocessing steps were carried out: normalising all upper case characters to lower case, removing stopwords, and filtering out all terms which occur in more than half of all documents. Stemmed index terms are obtained by applying the German or English Snowball stemmer (provided in the Lucene software) to topics and documents, For the retrieval experiments, the topic title and topic description were used as queries to Lucene.

While the Lucene software provides some support for decompounding in contributed modules, many changes were necessary to achieve the functionality required to conduct experiments on sub-word indexing. Decompounding words into CVC sequences was added as a new tokenizer generating multiple sub-words per word. For CVC sequences and n-grams (and variants), an additional word boundary marker was used (i.e. the character "|") at the beginning and end of a word. Lucene also provides a method to perform dictionary-based decompounding. Preliminary tests indicated that indexing with this method is very time-consuming (and will literally take days) due to inefficient lookup operations in the dictionary. Therefore, the dictionary representation in this method was changed from a set of words to a ternary search tree [Bentley and Sedgewick, 1997], which drastically improves indexing time. German and English (British English spelling) dictionaries were compiled from OpenOffice resources[6]. The German dictionary contains 133,379 entries, the English dictionary contains 46,280. The difference in the number of entries indicates the productivity of the German language to form new words as compounds.

For the hyphenation-based decompounding, hyphenation grammar files for German and English were provided by the Objects For Formatting Objects (OFFO) Sourceforge project.[7] Hyphenation points are inserted into words, defining syllable-like sub-words. Sub-words are required to have a minimum of 2 characters before and 2 characters after a hyphen, i.e. all sub-words have a minimum length of two characters. The character sequences between word boundaries and the hyphenation points are extracted as sub-words.

Time and disk space requirements for indexing and searching were calculated as the average number for two runs. The experiments were performed on a standard PC (Intel Core 2 Duo @ 3 GHz CPU, 4 GB memory, Western Digital 3200AAKS hard disk, OpenSuSe version 10.3 operating system).

5 Results and Discussion

Results for the German and English retrieval experiments are shown in Table 2. The following subsections describe retrieval performance, disk and time requirements, and a per-topic analysis of sub-word indexing.

5.1 Retrieval Performance

For German, with the exception of n-grams, all methods for indexing sub-words achieve a higher performance in comparison to stemming. The best performing sub-word indexing methods are to use CVC sequences and index them together with word stems (DE6: +17% MAP, +37% GMAP, +14% rel_ret), or to use syllable-like sub-words obtained from the hyphenation algorithm together with stems (DE12: +9% MAP, +23% GMAP, +11% rel_ret). Figure 3 shows the recall-precision graph for the experiments DE0, DE6, and DE12. The top five methods for German ordered by decreasing MAP are: CVC+stem, VCV+stem, HYPH+stem, DICT+stem, and stem.

An index comprising sub-words in some cases leads to a higher performance (e.g. DE5 vs. DE0, DE7 vs. DE0) compared to the baseline. An index with a combination of stopwords and stems always yields a higher performance compared to indexing sub-words only (e.g. DE2 vs. DE1, DE6 vs. DE5). Both recall (rel_ret) and precision (MAP, GMAP) are improved in the best experiments. In many cases, the number of relevant documents is higher than in the baseline (e.g. DE2, DE5, DE10, DE12). In most experiments, the initial precision (P@10, P@20) does not improve (e.g. DE13-DE18) or does not improve considerably (e.g. DE6 vs. DE0, DE12 vs. DE0).

The dictionary-based decompounding approach was expected to perform worse than approaches not requiring language-dependent or domain-specific resources, because the document corpus has a domain-specific vocabulary. Dictionary-based decompounding performs only slightly better than the baseline (e.g. DE10 vs. DE0).

Hyphenation was expected to outperform overlapping n-grams and CVC sequences, because the results correspond more to meaningful sub-words. Compared to CVC sequences, sub-words spanning word constituents are avoided by hyphenation, i.e. long consonant or vowel sequences spanning constituent words as in *"Geschäftsplan"* (business plan) or *"Seeigel"* (sea urchin) do not occur. Performance of the hyphenation is the second best for all methods (DE12) and clearly outperforms all n-gram methods.

Using overlapping character n-grams as indexing terms does not increase performance (DE13-DE18 and EN13-18). However, no combination of grams with different sizes was tried because combinations of other sub-words were not investigated in this paper (e.g. CV combined VC) and because of the additional disk space requirements.

The MAP for the experiments using CVC (DE6) and hyphenation-based sub-word indexing (DE12) is significantly higher than the MAP for the baseline experiment (Wilcoxon matched-pairs signed-ranks test, N=149, $p <= 0.0001$ and $p <= 0.05$ respectively).

For English, indexing sub-words does not outperform the baseline using standard retrieval on stemmed word forms (EN6: –8% MAP, –11% GMAP, +1% rel_ret for using CVC and stems). For two experiments, indexing CVC

[5] http://lucene.apache.org/

[6] http://wiki.services.openoffice.org/wiki/Dictionaries/

[7] http://offo.sourceforge.net/hyphenation/index.html

```
<DOC>
 <DOCID> GIRT-EN19900121783 </DOCID>
 <TITLE> Measures and projects of the Land of Lower Saxony for
  combatting female unemplyoment </TITLE>
 <AUTHOR> Wigbers, Antonia </AUTHOR>
 <PUBLICATION-YEAR> 1989 </PUBLICATION-YEAR>
 <LANGUAGE-CODE> EN </LANGUAGE-CODE>
 <COUNTRY-CODE> DEU </COUNTRY-CODE>
 <CONTROLLED-TERM> Lower Saxony </CONTROLLED-TERM>
 <CONTROLLED-TERM> woman </CONTROLLED-TERM>
 <CONTROLLED-TERM> employment promotion </CONTROLLED-TERM>
 <CONTROLLED-TERM> unemployment </CONTROLLED-TERM>
...
 <METHOD-TERM> documentation </METHOD-TERM>
 <METHOD-TERM> applied research </METHOD-TERM>
 <CLASSIFICATION-TEXT> Employment Research </CLASSIFICATION-TEXT>
</DOC>
```

Figure 1: Sample English GIRT4 document.

```
<top>
 <num> 177 </num>
 <EN-title> Unemployed youths without vocational training </EN-title>
 <EN-desc> Find publications focusing on jobless adolescents who have
  not completed any vocational training. </EN-desc>
 <EN-narr> Relevant documents give an overview of the scale and the
  problem of jobless adolescents who have not completed any job training.
  Not relevant are documents dealing exclusively with measures for youth
  welfare and youth policy. </EN-narr>
</top>
```

Figure 2: Sample English GIRT4 topic.

and hyphenated sub-words together with stems, the number of relevant and retrieved documents is slightly higher than in the baseline experiment (EN6 and EN12). No experiment improved MAP, GMAP or the precision at N documents in comparison to the baseline. The top five methods for English are: stem, CVC+stem, DICT+stem, HYPH+stem, and VCV+stem.

5.2 Disk space and time requirements

In addition to traditional information retrieval performance, the requirements to index and search the document collection using sub-word indexing were measured. More complex preprocessing requires more time, i.e. the time needed to process documents and queries increases.

All methods using a combination of stems and sub-words as indexing terms need more time and space than the baseline, which was an expected outcome. In the index combining stems and sub-words, all indexing terms from the baseline (stems) have to be generated and stored in addition to the sub-words. Dictionary-based decompounding requires the most additional time for indexing the document collection (+225.2% increase compared to the baseline). Hyphenation-based decompounding requires the most additional time for searching (+364.1%). However, a longer processing time is no guarantee for a better performance, as is shown by the dictionary-based approach.

5.3 Topic Analysis

The best two methods for decompounding (DE6 and DE12) were analysed in more detail on a per-topic basis. To obtain the average number of compounds in the topics, the following rules and guidelines for counting compound words were established:

- Abbreviated coordinations with hyphens count as one compound (e.g. *"Parlaments- oder Präsidentschaftswahlen"*).
- Words with bound morphemes count as a compound (e.g. *"Kneipengänger"*).
- Words with non-separable prefixes count as a compound (e.g. *"Ökosteuer"*).
- Hyphenated words do not count as compound words (e.g. *"burn-out"*).
- Compounds are not limited to nouns, but also include verbs and adjectives (e.g. *"rechtsextrem"*).
- Words which may be incorrectly decomposed into constituent words do not count as compounds (e.g. *"Mutterschaft"*).

Following these guidelines, the GIRT topics were manually annotated. The 150 topics contain an average of 1.49 compounds per topic. The topics for the best-performing methods were sorted by gain in MAP. For CVC (DE6), the average number of compounds in the top-20 topics is 2.15, for HYPH (DE12) the average is 2.3. There is an overlap of 17 topics of the top-20 best performing topics for experiments DE6 and DE12.

There are two topics among the top-20 which do not contain any compounds at all, topic 82: *"Berufliche Bildung von Immigranten"* (Professional training of immigrants)/ *"Finde Dokumente, die über die berufliche Integration von Immigranten durch berufliche Bildung berichten"* (Find

Table 2: Results for monolingual retrieval experiments on German and English GIRT4 documents (lang.: language; rel_ret: number of relevant and retrieved documents).

Run	Parameters	Results							
ID	index terms	rel_ret	MAP	GMAP	P@10	P@20	indexing [s]	searching [s]	index size [MB]
DE0	stem	11025	0.3214	0.2097	0.63	0.55	279.5	**17.3**	**659**
DE1	CV	10778	0.2715	0.1554	0.52	0.46	338.8 (+21.2%)	32.5 (+87.8%)	1106 (+67.8%)
DE2	CV+stem	12108	0.3494	0.2537	0.62	0.55	576.6 (+106.2%)	40.9 (+136.4%)	1412 (+114.2%)
DE3	VC	10480	0.2399	0.1308	0.47	0.41	339.7 (+21.5%)	68.4 (+295.3%)	1075 (+63.1%)
DE4	VC+stem	11819	0.3317	0.2448	0.60	0.54	532.5 (+90.5%)	43.3 (+150.2%)	1383 (+109.8%)
DE5	CVC	12360	0.3584	0.2673	0.63	0.56	472.6 (+69.0%)	52.7 (+204.6%)	1285 (+94.9%)
DE6	CVC+stem	**12599**	**0.3765**	**0.2886**	**0.65**	**0.58**	631.8 (+126.0%)	39.0 (+125.4%)	1585 (+140.5%)
DE7	VCV	11879	0.3311	0.2309	0.59	0.53	358.7 (+28.3%)	37.2 (+115.0%)	1185 (+79.8%)
DE8	VCV+stem	12477	0.3654	0.2771	0.63	0.56	729.5 (+161.-%)	49.6 (+186.7%)	1492 (+126.4%)
DE9	DICT	11545	0.3051	0.1958	0.53	0.49	617.2 (+120.8%)	63.8 (+268.7%)	1170 (+77.5%)
DE10	DICT+stem	12252	0.3450	0.2447	0.61	0.53	909.2 (+225.2%)	75.0 (+333.5%)	1376 (+108.8)
DE11	HYPH	11743	0.3217	0.2269	0.59	0.53	433.5 (+55.0%)	40.4 (+133.5%)	896 (+35.9%)
DE12	HYPH+stem	12291	0.3511	0.2582	0.62	0.56	682.0 (+144.0%)	80.3 (+364.1%)	1111 (+68.5%)
DE13	3-gram	10380	0.2518	0.1546	0.51	0.45	473.6 (+69.4%)	67.3 (+289.0%)	1582 (+140.0%)
DE14	3-gram+stem	10901	0.2835	0.1940	0.54	0.50	774.7 (+177.1%)	70.8 (+309.2%)	1809 (+174.5%)
DE15	4-gram	9961	0.2429	0.1590	0.52	0.47	376.3 (+34.6%)	51.1 (+195.3%)	1338 (+103.0%)
DE16	4-gram+stem	10180	0.2547	0.1716	0.54	0.48	633.8 (+126.7%)	54.2 (+213.2%)	1503 (+128.0%)
DE17	5-gram	7824	0.1765	0.0911	0.48	0.41	**277.5 (-0.8%)**	29.5 (+70.5%)	964 (+46.2%)
DE18	5-gram+stem	8095	0.1876	0.1017	0.50	0.43	352.5 (+26.1%)	48.1 (+178.3%)	1058 (+60,.5%)
EN0	stem	**10911**	**0.3453**	**0.2239**	**0.57**	**0.53**	179.6	**12.0**	**275**
EN1	CV	9027	0.2144	0.1049	0.43	0.38	171.3 (-4.7%)	25.0 (+108.3%)	493 (+79.2%)
EN2	CV+stem	10573	0.3002	0.1804	0.54	0.48	268.9 (+49.7%)	32.0 (+166.6%)	626 (+127.6%)
EN3	VC	8576	0.1800	0.0797	0.38	0.34	174.5 (-2.9%)	23.8 (+98.3%)	483 (+75.6%)
EN4	VC+stem	10551	0.2953	0.1802	0.54	0.48	265.2 (+47.6%)	29.4 (+145.0%)	615 (+123.6%)
EN5	CVC	10545	0.2929	0.1775	0.55	0.48	186.9 (+4.0%)	25.9 (+115.8%)	551 (+100.3%)
EN6	CVC+stem	10985	0.3181	0.1993	0.56	0.50	304.8 (+69.7%)	30.9 (+157.5%)	679 (+146.9%)
EN7	VCV	10082	0.2649	0.1557	0.51	0.45	189.0 (+5.2%)	30.8 (+156.6%)	526 (+91.2%)
EN8	VCV+stem	10759	0.3074	0.1952	0.56	0.50	255.6 (+42.3%)	30.1 (+150.8%)	658 (+139.2%)
EN9	DICT	10163	0.2797	0.1587	0.53	0.47	281.9 (+56.9%)	38.1 (+217.5%)	561 (+104.0%)
EN10	DICT+stem	10785	0.3139	0.1915	0.55	0.50	390.7 (+117.5%)	41.9 (+249.1%)	640 (+132.7%)
EN11	HYPH	10451	0.2813	0.1740	0.53	0.46	206.4 (+114.9%)	23.4 (+95.0%)	376 (+36.7%)
EN12	HYPH+stem	10908	0.3104	0.1944	0.53	0.48	303.7 (+69.0%)	28.1 (+134.1%)	460 (+67.2%)
EN13	3-gram	9549	0.2388	0.1410	0.49	0.43	228.3 (+27.1%)	43.9 (+265.8%)	712 (+158.9%)
EN14	3-gram+stem	9989	0.2678	0.1668	0.53	0.47	295.3 (+64.4%)	48.3 (+302.5%)	831 (+202.1%)
EN15	4-gram	8709	0.2149	0.1128	0.47	0.41	173.6 (-3.4%)	22.2 (+85.0%)	573 (108.3%)
EN16	4-gram+stem	8964	0.2317	0.1238	0.50	0.44	260.6 (+45.1%)	27.6 (+130.0%)	663 (+141.0%)
EN17	5-gram	6236	0.1482	0.0611	0.42	0.35	**146.2 (-18.6%)**	15.4 (+28.3%)	388 (+41.0%)
EN18	5-gram+stem	6354	0.1535	0.0660	0.43	0.36	207.6 (+15.5%)	16.0 (+33.3%)	439 (+59.6%)

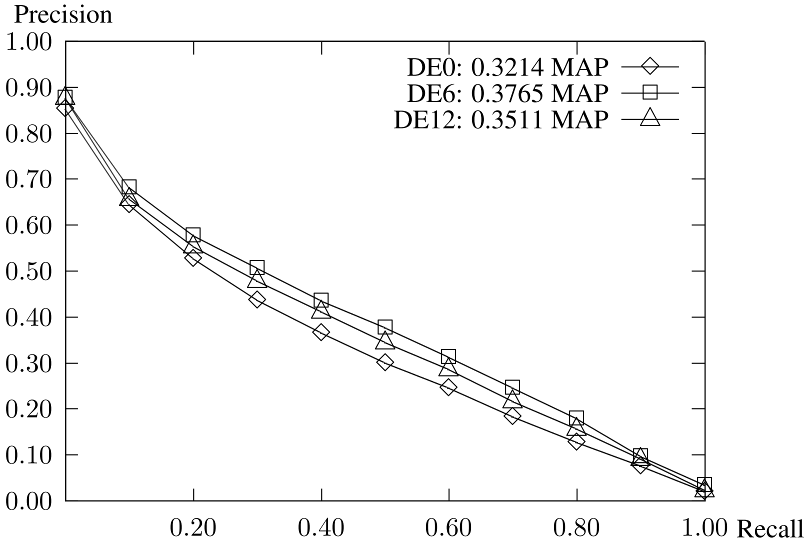


Figure 3: Recall-precision graph for selected experiments.

documents on the professional integration of immigrants through vocational training) and topic 101: *"Tiere in der Therapie"* (Animals in therapy)/ *"Finde Dokumente, die über das Nutzen des Potenzials von Tieren in der therapeutischen Arbeit mit dem Menschen berichten"* (Find documents reporting on the potential of using animals in human therapeutic programs).

In topic 82, standard IR methods like stemming do not allow matching possibly relevant documents mentioning *"Immigration"* (immigration) instead of *"Immigranten"* (immigrants). If decompounding is used to split words into sub-words, these different but semantically related words will have some sub-words in common and additional documents can be found.

For topic 101, the terms *"Therapie"* (therapy) and *"therapeutisch"* (therapeutical) are usually stemmed to different indexing terms and each will have a low weight assigned to them. Using sub-words, these word forms share some sub-words assigned to them and the shared sub-words will have a higher weight. In addition, this topic contains a word with the new German spelling, *"Potenzial"* (potential)). Most documents in the GIRT collection were written before the spelling was changed. The term *"Potential"* in the old German spelling has a term frequency of 2323, *"Potenzial"* has a frequency of 76. Thus, very few documents containing the new spelling will be found. Matching terms on a sub-word level (instead of exact matching on the word-level) will yield more potentially relevant documents.

5.4 Summary

In summary, sub-word indexing does not perform equally for the non-compounding language English in comparison to the compounding language German. Most German experiments clearly outperform the stemming baseline with respect to retrieval metrics MAP, GMAP, P@10, and P@20.

All sub-word indexing methods require more time for indexing and searching a database. In addition, the index size for sub-words is higher compared to a stem index. The size of a combined index (using sub-words and stems as indexing units) is up to an additional 174% of the original size.

Indexing time for 5-grams is lower than the indexing time for the stemming baseline. The required time to index and search a collection increases with the number of indexing units produced. In a combined index (sub-words and stems), the stems also have to be produced. Additionally, typically several sub-words are identified for each word. Thus, indexing and searching sub-words requires more time than for the stemming baseline.

The best performing methods – CVC indexing and hyphenation-based sub-word indexing – perform significantly better than the stemming baseline for German, they perform best on very similar topics, and they even improve some topics which do not contain compounds at all.

6 Conclusion and Future Work

Four different approaches to break up words for indexing sub-words were discussed and evaluated on the German and English data for the domain-specific track GIRT at CLEF. Three of the methods outperform the stemming baseline. These methods include consonant-vowel sequences, which have been mostly used for spoken document retrieval and a new method for decompounding, based on hyphenation patterns to find sub-words. In comparison to the standard stemming baseline, decompounding yields a significantly higher performance in terms of MAP, GMAP, and rel_ret for German. In conclusion, sub-word indexing for German may be seen as a method integrating decompounding and stemming: words are broken down into smaller indexing units and frequent affixes are either removed completely or are associated with a low weight.

The best performing methods are also very cost-effective and easily adaptable to other languages. Consonant-vowel sequences can be produced as a by-product of stemming and stemmers already exist for many languages. Snowball contains stemmers for about 16 languages. Similarly, there already are TeX hyphenation rules for more than 30 different languages as well. Indexing n-grams did not produce results comparable to or higher than the stemming baseline. For English, sub-word indexing does not perform as good as stemming, most likely because English words do not have to be split into smaller units.

Splitting compounds into several smaller indexing units considerably changes many implicit parameters for IR, including the number of terms in both queries and documents, term frequencies, and the average document length. These changes suggest that parameters should be adjusted and optimised correspondingly if a different weighting model is applied. Future work will include experiments with state-of-the-art retrieval models (e.g. OKAPI BM25, [Robertson *et al.*, 1994]), determining parameters based on the new characteristics of the index and topics. The effect of sub-words on relevance feedback will be investigated for different sub-word indexing methods.

Acknowledgments

This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (CNGL) project.

References

[Bentley and Sedgewick, 1997] Jon L. Bentley and Robert Sedgewick. Fast algorithms for sorting and searching strings. In *SODA '97: Proceedings of the eighth annual ACM-SIAM symposium on discrete algorithms*, pages 360–369, Philadelphia, PA, USA, 1997. Society for Industrial and Applied Mathematics.

[Braschler and Ripplinger, 2003] Martin Braschler and Bärbel Ripplinger. Stemming and decompounding for German text retrieval. In F. Sebastiani, editor, *ECIR 2003*, volume 2633 of *Lecture Notes in Computer Science (LNCS)*, pages 177–192, Berlin, 2003. Springer.

[Chen and Gey, 2004] Aitao Chen and Fredric C. Gey. Multilingual information retrieval using machine translation, relevance feedback and decompounding. *Information Retrieval*, 7(1–2):149–182, 2004.

[Chen, 2003] Aitao Chen. Cross-language retrieval experiments at CLEF 2002. In Carol Peters, Martin Braschler, Julio Gonzalo, and Michael Kluck, editors, *Advances in Cross-Language Information Retrieval, Third Workshop of the Cross-Language Evaluation Forum, CLEF 2002. Rome, Italy, September 19-20, 2002. Revised Papers*, volume 2785 of *Lecture Notes in Computer Science (LNCS)*, pages 28–48. Springer, Berlin, 2003.

[Daumke *et al.*, 2007] Philipp Daumke, Jan Paetzold, and Kornel Marko. MorphoSaurus in ImageCLEF 2006: The effect of subwords on biomedical IR. In Carol Peters, Paul Clough, Fredric C. Gey, Jussi Karlgren, Bernardo Magnini, Douglas W. Oard, Maarten de Rijke, and Maximilian Stempfhuber, editors, *Evaluation of Multilingual and Multi-modal Information Retrieval: 7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006, Alicante, Spain, Revised Selected Papers*, volume 4730 of *Lecture Notes in Computer Science (LNCS)*, pages 652–659. Springer, Berlin, 2007.

[Daumke, 2007] Philipp Daumke. *Das MorphoSaurus-System – Lösungen fur die linguistischen Herausforderungen des Information Retrieval in der Medizin*. PhD thesis, Albert-Ludwigs-Universität, Freiburg i.Br., Medizinische Fakultät, 2007.

[Glavitsch and Schäuble, 1992] Ulrike Glavitsch and Peter Schäuble. A system for retrieving speech documents. In *Proceedings of ACM SIGIR 1992*, pages 168–176, Denmark, 1992.

[Kamps *et al.*, 2003] Jaap Kamps, Christof Monz, and Maarten de Rijke. Combining evidence for cross-language information retrieval. In Carol Peters, Martin Braschler, Julio Gonzalo, and Michael Kluck, editors, *Advances in Cross-Language Information Retrieval, Third Workshop of the Cross-Language Evaluation Forum, CLEF 2002. Rome, Italy, September 19-20, 2002. Revised Papers*, volume 2785 of *Lecture Notes in Computer Science (LNCS)*, pages 111–126. Springer, Berlin, 2003.

[Kluck, 2005] Michael Kluck. The domain-specific track in CLEF 2004: Overview of the results and remarks on the assessment process. In Carol Peters, Paul Clough, Julio Gonzalo, Gareth J. F. Jones, Michael Kluck, and Bernardo Magnini, editors, *Multilingual Information Access for Text, Speech and Images, 5th Workshop of the Cross-Language Evaluation Forum, CLEF 2004, Bath, UK, September 15-17, 2004, Revised Selected Papers*, volume 3491 of *Lecture Notes in Computer Science (LNCS)*, pages 260–270. Springer, Berlin, 2005.

[Knuth, 1984] Donald E. Knuth. *Computers & Typesetting. Volume A. The TeXbook*. Addison-Wesley, Reading, Mass., 1984.

[Liang, 1983] Franklin Mark Liang. *Word hy-phen-a-tion by com-put-er*. PhD thesis, Stanford University, Department of computer science, Stanford, CA, USA, 1983.

[McNamee and Mayfield, 2007] Paul McNamee and James Mayfield. N-gram morphemes for retrieval. In *Working Notes of the CLEF 2007 Workshop*, Budapest, Hungary, September 2007.

[McNamee, 2001] Paul McNamee. Knowledge-light Asian language text retrieval at the NTCIR-3 workshop. In Keizo Oyama, Emi Ishida, and Noriko Kando, editors, *Proceedings of the Third NTCIR Workshop on Research in Information Retrieval, Automatic Text Summarization and Question Answering*, Tokyo, Japan, 2001. National Institute of Informatics (NII).

[Ng, 2000] Kenney Ng. *Subword-based approaches for spoken document retrieval*. PhD thesis, Massachusetts institute of technology (MIT), Department of electrical engineering and computer science, 2000.

[Porter, 1980] Martin F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[Robertson *et al.*, 1994] Stephen E. Robertson, Steve Walker, Susan Jones, and Micheline Hancock-Beaulieu. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference (TREC 1994)*, Gaithersburg, USA, 1994.

[Schäuble and Glavitsch, 1994] Peter Schäuble and Ulrike Glavitsch. Assessing the retrieval effectiveness of a speech retrieval system by simulating recognition errors. In *HLT '94: Proceedings of the workshop on Human Language Technology*, pages 370–372, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

Multi-facet Classification of E-Mails in a Helpdesk Scenario

Thomas Beckers
Universität Duisburg-Essen
Dep. of Computer Science
47048 Duisburg, Germany
tbeckers@is.inf.uni-due.de

Ingo Frommholz
University of Glasgow
Dep. of Computing Science
G12 8QQ Glasgow, UK
ingo@dcs.gla.ac.uk

Ralf Bönning
d.velop AG
Schildarpstraße 6 – 8
48712 Gescher, Germany
ralf.boenning@d-velop.de

Abstract

Helpdesks have to manage a huge amount of support requests which are usually submitted via e-mail. In order to be assigned to experts efficiently, incoming e-mails have to be classified w. r. t. several facets, in particular topic, support type and priority. It is desirable to perform these classifications automatically. We report on experiments using Support Vector Machines and k-Nearest-Neighbours, respectively, for the given multi-facet classification task. The challenge is to define suitable features for each facet. Our results suggest that improvements can be gained for all facets, and they also reveal which features are promising for a particular facet.

1 Introduction

The impact of e-mail for business communication has grown dramatically during the last years. These e-mails have often a context in a business workflow. They may be trigger events for the start of a business process like an order request or they may be parts of knowledge intensive tasks [Abecker *et al.*, 2000] [Frommholz and Fuhr, 2006]. In this paper a case study of multi-facet e-mail classification for the helpdesk scenario of the d.velop AG is given. One major difficulty in e-mail classification research is the availability of data sets with correlations to the business workflow context. Although with the Enron data set [Klimt and Yang, 2004] a set of e-mails of a real world company is given, these e-mails have no explicitly given context in a business process.

To allow for the immediate dissemination of an incoming e-mail to an appropriate agent, it has to be classified w. r. t. the following three facets. A *topical classification* is necessary to determine what an e-mail is about and to find the right expert for it. Choosing a wrong person for a specific request results in additional waiting time for the customer. This may be crucial for high priority calls. The *type* of an e-mail is another important criterion – while actual support requests must be assigned to an expert, e-mails containing, for instance, criticism or a few words of gratitude, but no support request, may not be distributed at all in order to keep an expert from extra work. The third important facet is the *priority* of an e-mail, which is useful either for selecting the right expert (e. g. , someone who is immediately available in case of high priority) on the one hand, and for giving the associated expert a hint whether the request has to be handled immediately or not on the other hand. Service Level Agreements (SLA) exist that define response times for different priority categories.

The problem we are dealing with is thus a multi-facet classification of e-mails w .r. t. the three facets described above. While topical classification is a well-understood problem, classification w .r. t. the other two non-topical facets is a challenging and novel task.

The remainder of the paper is structured as follows. First, we discuss some related work on e-mail classification. Subsequently, we introduce the collection we are dealing with and discuss the facets in more detail. The methods and features used for multi-facet classification are presented in section 4. Section 5 shows some evaluation and discusses the results. Finally, a conclusion and an outlook on future work are given in section 6.

2 Related Work

E-mails are one of the most frequently used services of the internet. They are specified by *RFC 2822* of the Internet Engineering Task Force (IETF). E-mails can be considered as semi-structured documents. Semi-structured means that there is no full underlying data model as it is common in databases. Though, certain parts are described by a model, like the date or the content type, while other parts have no structure at all, like the body containing the actual message.

Most research focuses on the classification into an existing folder structure created by the user [Koprinska *et al.*, 2007] [Bekkerman *et al.*, 2004] [Crawford *et al.*, 2004] [Brutlag and Meek, 2000] [Rennie, 2000] [Segal and Kephart, 1999]. This folder structure is usually of topical nature, that is, a folder contains e-mails which are about the same topic. One main challenge is the continuous adding and removing of e-mails from folders. In contrast to most other classification tasks, one has to deal with dynamic classes. That is why this process is sometimes referred to as *filtering*. Koprinska et al. achieved the best results for topical folder structures. Eichler [2005] classified e-mails of a Swedish furniture retailer with regard to the classes *assortment*, *inventory* and *complaint* but only a few selected e-mails were used.

The most common application of e-mail classification in daily use is certainly the classification of unwanted e-mails (spam). Many researchers have introduced their concepts. By now it is possible to achieve an accuracy of about 90%. Blanzieri and Bryl [2006] as well as Cormack [2007] provide a survey about current techniques for the classification of spam e-mails.

Classification in regard to non-topical criteria is also possible. Cohen et al. [2004] classified e-mails according to so-called *acts of speech*. An act of speech is a pair of a verb and a noun, like *deliver information* or *request meeting*. Bennett and Carbonell [2005] tried to recognize e-mails that require an action by the user (*action items*). Their classification was performed on document and sentence level whereas classification on sentence level achieved the best results. Nenkova and Bagga [2003] analysed e-mails of a contact center if they require an immediate reply or not (*root messages* vs. *single messages*). Antoniol et al. [Antoniol *et al.*, 2008] classified texts posted in bug tracking systems – which are similar to e-mails – into different kind of activities, like *bug*, *enhancement*, *refactoring* etc.

Most research focuses on term features only and performs classification with respect to a single facet. Furthermore, only corpora in English are used. Our approach takes also non-term features into account and is evaluated with a German language corpus.

3 Collection

The d.velop AG[1] is a German software company for solutions and systems in the area of document management. In order to give their customers support on their products the support department has implemented a helpdesk solution in which customer requests are stored, tracked and routed to appropriate agents.

The d.velop AG has made a subset of support e-mails – so-called *tickets* – available for this research work. One motivation for this step is the question whether the helpdesk process can be made more efficient with the support of automatic classification. Furthermore, the d.velop AG has developed commercial solutions for managing e-mails and aims at improving their products based on research work.

Our collection consists of 2000 e-mails that were received from October 2007 to May 2008 by the support system at d.velop AG. A multi-facet classification was performed by the employees of the helpdesk which we used for training and testing of classifiers.

Every incoming e-mail is stored in two files by the support system. One file contains the actual content while the other file contains metadata. The metadata includes the classifications in respect to the three facets which were performed by the employees of the helpdesk. This classfication is used for the training of the classifiers. In order to handle the data more easily we created a single XML file which comprises all relevant data. Furthermore, we cleaned up the classes and deleted tickets internal to d.velop. Some rare classes were also removed while some other classes were merged.

In the following the classes of each facet are described.

Topic Each product or service forms a class. A class consists of three parts, namely the *product*, the *module* and the *component*. The *module* and the *component* part can also be empty. Thus, the classes build several hierarchies. Figure 1 illustrates an extract of the class hierarchies.

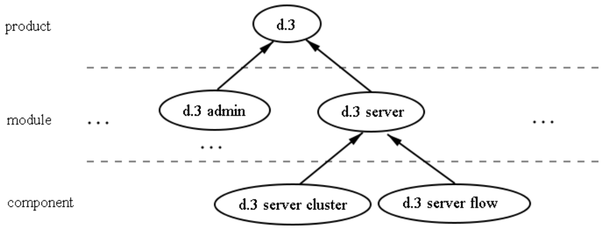


Figure 1: Extract of the class hierarchies of the facet *topic*

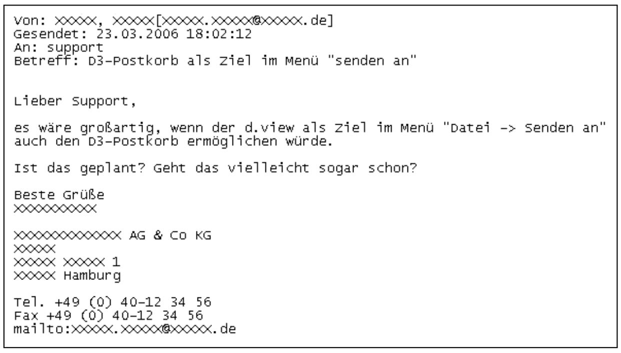
Some classes occur quite often, such as *d.3 explorer*, while others occur very rarely, like *d.3 admin ldap*. 31 classes remained after cleanup.

Support Type The classes of the support type are

- *error report*, if an error occurs in the software (merged from 2 very similar classes);
- *questions concerning handling/config/doc*, if there are questions about the handling, configuration or documentation of the software;
- *consultation*, if a customer requests consultation about products or services and
- *miscellaneous*, for all other possible types of requests.

Three other classes were ignored since they were rarely used. About 70% of the tickets belong to the class *error report*. Note that a classification as error does not make a statement about the severity or if it is a product error at all.

Priority A ticket can be classified as *urgent* if it requires an immediate response, like an error report about a problem that crashes a complete system. Otherwise, a ticket is classified as *not urgent*. Both classes comprise two of the original four priority classes. Most tickets are *not urgent*.



```
Von: XXXXX, XXXXX[XXXXX.XXXXX@XXXXX.de]
Gesendet: 23.03.2006 18:02:12
An: support
Betreff: D3-Postkorb als Ziel im Menü "senden an"


Lieber Support,

es wäre großartig, wenn der d.view als Ziel im Menü "Datei -> Senden an"
auch den D3-Postkorb ermöglichen würde.

Ist das geplant? Geht das vielleicht sogar schon?

Beste Grüße
XXXXXXXXXX

XXXXXXXXXXXXX AG & Co KG
XXXXX
XXXXX XXXXX 1
XXXXX Hamburg

Tel. +49 (0) 40-12 34 56
Fax +49 (0) 40-12 34 56
mailto:XXXXX.XXXXX@XXXXX.de
```

Figure 2: Example of a ticket that was received via e-mail

Fig. 2 shows an example of a ticket[2] received via e-mail. A customer asks if it is possible to add a new shortcut to a menu. With regard to the facets this ticket is classified as

- *d.view* (facet *topic*),
- *question concerning handling/config/doc* (facet *support type*) and
- *not urgent* (facet *priority*).

[1] `http://www.d-velop.de/en/`

[2] Some details were made irrecognisable because of privacy reasons. Telephone numbers are fictitious.

4 Multi-facet Classification

Classification was performed with Support Vector Machines (SVMs). Due to the fact that most text classification problems are linearly separable [Joachims, 1998] a linear kernel was employed. SVMs are used with a classic representation of documents, but including also non-term features besides term features. That is why this technique is called *extended indexing* in the following.

Alternatively, we utilised k-Nearest-Neighbour (k-NN) as classification techniques. For k-NN we made use of a *probabilistic, decision-oriented indexing* technique developed by Fuhr and Buckley [1991]. Features of terms and documents (tickets) x are defined as

$$\vec{x}(t,d) = (x_1(t,d), x_2(t,d), \ldots, x_3(t,d)),$$

whereas t denotes a term and d denotes a document. For example, $\vec{x}(t,d)$ could be defined as

$$\begin{aligned} x_1(t,d) &= \begin{cases} 1 & \text{if } t \text{ in } d \text{ occurs once} \\ 2 & \text{if } t \text{ in } d \text{ occurs at least twice} \end{cases} \\ x_2(t,d) &= idf(t) \\ x_3(t,d) &= \begin{cases} 1 & \text{if } t \text{ occurs in the subject of } d \\ 0 & \text{else} \end{cases}, \end{aligned}$$

with $idf(t)$ as the inverse document frequency of t. These features are used to learn an indexing function which estimates $P(R|\vec{x}(t,d))$ based on a learning sample L^x. Beckers [2008] shows in detail how L^x is constructed. This probability is then used as indexing weight for term t in document d. The terms of the tickets and their weights are used as features for classification with k-NN. Logistic regression based on a maximum-likelihood criterion was employed to learn the indexing function. Our approach is similar to that of Gövert et al. [1999], who classified web documents of Yahoo's web catalogue.

After the representations of the tickets have been created, normalisation of the data was applied. SVMs as well as k-NN require normalisation since features with large values would otherwise overlie features with small values. The preparation of the collection and the features are outlined in the following.

4.1 Features

We regard term features as well as non-term features for classification. We defined features which seem to be useful for our task. Features and groups of features, respectively, are divided into feature categories. For each of our three facets all features are regarded. We defined the following feature categories for the extended indexing.

Terms The most obvious features are the terms which appear in the tickets. They can be either represented as sets of words or the frequency of their occurrence can be regarded. Another possibility is not to consider all terms but only terms from a dictionary (*special terms*). N-grams are usually used to take the context of terms into account. We only use bigrams because n-grams with $n > 2$ excessively increase the dimensionality of the data. Thus, most n-grams would only occur rarely or once. Finally, there are some statistics features; the count of the number of terms and the number of different terms.

Term position Not only the terms can provide meaningful features for classification. A term can appear in certain fields, like the subject or the attachment and at different places of the body. Thus, the body is divided into three thirds. Also, a simple recognition of interrogative sentences is performed. A suffix representing the position is appended to each term. These terms plus suffix are used as features.

Punctuation The usage of punctuation may also be useful for classification [Nenkova and Bagga, 2003]. Consider the following sentence of a ticket: "This does not work!". An exclamation mark may be used more often in problem reports than in questions concerning the documentation. Thus, there are features about the usage (number, relative number and if there are three in a row) of exclamation and question marks.

Attachment The attachment is used to create features as well. The actual content of the attached files is ignored since it is nearly impossible to extract data from all possible attachment types. If there are attachments and the types thereof are regarded. There are binary features for each of the following file types:

- log files (*.log)
- text files (*.txt)
- XML files (*.xml)
- temporary files (*.tmp, *.temp)
- images (*.jpg, *.png, *.bmp, *.tif, *.gif, ...)
- archives (*.zip, *.jar, *.rar)
- miscellaneous

Sender The sender address is used as feature in its entirety. The domain part of the address is used as another feature. There is also some historical information about the classes of past tickets.

Length The length of the subject and the length of the body are two more features. Note that these both features count the character length while the length feature from the feature category terms counts the terms.

Time The date and the time of an incoming ticket is also of potential value for classification. We use several features of time. There are 7 binary features that indicate the day of the week. The 24 hours of a day are divided into several blocks. For each of these blocks there is also a binary feature. Finally, a binary feature shows if the time is during usual labour time.

Characters Problem reports often have inlined snippets of e. g. log files or error messages which contain many special characters. Some special characters that we regard are e. g. the following:

```
- [ ] ( ) { } : _ + = # * $ & % / \ ~ | @
```

An overview about the features described above can be found in tables 10 and 11 in the appendix.

The probabilistic, decision-oriented indexing requires different definitions of term features and thus different defined feature categories (see sec. 4). All other non term-related feature categories stay the same. These features are used to learn an indexing function with logistic regression. The indexing function is then used to compute the weights for the terms of the tickets.

Terms The term frequency and the inverse document frequency of a term build this feature category as well as a binary feature that checks if a terms belongs to the most frequent terms. The statistics-related features are defined as stated above.

Term position All features from this feature category are defined along the lines of the feature category of the extended indexing but they corresponded to terms instead of tickets.

A more detailed description of the features as well as additional examples are provided by Beckers [2008].

5 Evaluation

We used the classic 10-fold stratified cross validation for testing of our concepts. Our main questions are:

- Is it possible to classify with higher quality compared to a given baseline?
- How do the different approaches (ext. indexing & SVM and prob. indexing and k-NN) perform?
- Which features are appropriate for the different facets? We think that not only set of words should be regarded as features, especially for non-topical facets.

In the following, we describe the implementation, the creation of the training/test collection, the selection of appropriate evaluation measures and finally the achieved results.

5.1 Implementation

Our experiments were performed with the open-source data mining framework *RapidMiner*[3]. Additional functionality has been implemented by means of *RapidMiner*'s plug-in mechanism. For classification with SVMs we selected the LibSVM operator which wraps the well-known LIBSVM[4] library. A built-in operator was used for k-NN. We applied *RapidMiner* in version 4.2. All experiments ran on a computer with AMD Phenom 9550 2.2 GHz, 4 GB RAM, Debian 2.6.18.gfsg-1-22 (Xen) and Java JDK 6 Update 7 64 bit.

5.2 Training and Test Collection per Facet

The complete collection consists of 2000 tickets. The maximum number of available tickets is used for the facet *topic*. Tickets with rare classes were removed, that is, classes that only occur less than 15 times. This results in 1643 tickets usable for classification. Due to time constraints only 1000 arbitrary tickets were selected for the facet *support type*. As there are only four different classes a smaller number of tickets is likely to be sufficient. Because of the poor quality of the classification for the facet *priority* we used 150 manually selected tickets for each class.

5.3 Evaluation Measures

Due to the different nature of the facets, a single evaluation measure for all facets would not have been appropriate. It seems reasonable to define a cost-based measure which takes the hierarchical structure of the classes from the facet *topic* in to account. $cost_i^j$ denotes the costs of classifying an instance of class c_j as instance of class c_i. The classification costs are defined as

$$cost = \frac{1}{|Test|} \cdot \sum_{t \in Test} cost_{predictedClass(t)}^{actualClass(t)},$$

whereas $actualClass(t)$ and $predictedClass(t)$ denote the index of the actual and the predicted class, respectively. $Test$ denotes the set of test instances.

	costs
$c_{predicted}$ equals c_{actual}	0
$c_{predicted}$ is ancestor of $c_{actual}(2 \rightarrow 3)$	0.3
$c_{predicted}$ is ancestor of $c_{actual}(1 \rightarrow 2)$	0.3
$c_{predicted}$ is ancestor of $c_{actual}(1 \rightarrow 3)$	0.7
c_{actual} is ancestor of $c_{predicted}(2 \rightarrow 3)$	0.3
c_{actual} is ancestor of $c_{predicted}(1 \rightarrow 2)$	0.3
c_{actual} is ancestor of $c_{predicted}(1 \rightarrow 3)$	0.7
$c_{predicted}$ and c_{actual} are siblings (3)	0.3
$c_{predicted}$ and c_{actual} are siblings (2)	0.7
otherwise	1

Table 1: The classification costs of a ticket of class c_{actual} that was predicted as class $c_{predicted}$ for the *topic* facet

Table 1 shows the costs that are heuristically defined based on experience. The numbers in brackets denote the levels in the hierarchy. A correct classification does not produce any costs while a classification that is completely wrong produces maximum costs. If a predicted class is an ancestor or a sibling the costs are somewhere in between.

A ticket of class *d.view* (see fig. 2) which gets classified as *d.view admin* would cause costs of 0.3.

For the facet *support type* the well-known accuracy was used [Sebastiani, 2002]. There's no evidence that it makes sense to assign different costs for wrongly classified tickets. The accuracy a is defined as

$$a = \frac{TP + TN}{TP + TN + FP + FN},$$

with the usual definition of TP (true positive), FP (false positive), TN (true negative) and FN (false negative).

Cost matrix for priority		**predicted class**	
		urgent	not urgent
actual class	urgent	0	2
	not urgent	1	0

Table 2: Cost matrix for facet *priority*

The facet *priority* also uses a cost-based measure (see table 2). It is worse to classify an urgent ticket as not urgent than classifying a not urgent ticket as urgent. It is important that the processing of urgent

[3] http://sourceforge.net/projects/yale/

[4] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

tickets is not delayed by an erroneous classification. That is why the costs are twice as much for the former case as for the latter case.

5.4 Results

First, we examined the term features including term modification (stemming, stop word removal, removal of rare terms) in detail. If feature categories have shown to be useful they were kept for the following experiments. Afterwards, the non-term features were analyzed. We performed an experiment that covered all non-term feature categories and then for each non-term feature category an experiment without it was performed in order to determine its usefulness. Finally, all useful term and non-term features were regarded. The optimal parameters for SVM and k-NN were estimated after each step, namely C and k, respectively. To handle multi-class problems with SVMs we utilised the one vs. one approach [Hsu and Lin, 2002]. Beckers [2008] provides a more detailed description of the results and computed additional evaluation measures for the best results.

Baselines

We regarded two different baselines, namely a random classification and a classification taking the most common class (mode). Table 3 shows the baselines for each facet. Note that costs are the better the smaller they are while the accuracy should be as high as possible. Based on these baselines we perform t-tests with $\alpha = 0.05$ (☆) and $\alpha = 0.01$ (★). There is no mode baseline for the *priority* (see sec. 5.2).

	mode	**random**	**measure**
topic	0.7117	0.8613	costs
support type	0.7179	0.2393 0.2419[1]	accuracy
priority	–	0.7334	costs

[1] All instances were weighted inversely proportional with the occurrence frequency of their class.

Table 3: Results of the baseline experiments

Facet *Topic*

Ext. Indexing & SVM Table 4 shows the results of the experiments for this facet. The best result (0.3954) of SVMs was achieved by applying simple term features with binary weights (set of words) and term modification (printed in bold font). Only special terms as feature also achieved good results (row 4) but with a lower dimensionality of the data and thus with increased performance. So, if a slight decrease of classification quality is acceptable, then a significant faster learning of classifiers is possible. Bigrams apparently were not appropriate. The term position features were also of some value for classification. All non-term features except sender and character features provided useful information for classification. Using both term and non-term features could not increase the classification quality. All results are statistically significant w. r. t. both baselines. As expected term features are the most useful features. Non-term features decreased the costs below the baselines but they could not improve the overall classification quality.

experiment	**costs**	**SM**[1]	**SR**[2]
terms (binary)	0.4212	★	★
terms (binary & mod.)	**0.3954**	★	★
terms (tf)	0.5082	★	★
terms (special terms)	0.4154	★	★
terms (bigrams)	0.5424	★	★
terms	0.3957	★	★
term position	0.4454	★	★
all non-term features	0.6359	★	★
without punctuation	0.6362	★	★
without attachment	0.6779	★	★
without sender	0.6252	★	★
without length	0.636	★	★
without time	0.6363	★	★
without characters	0.6357	★	★
all	0.3991	★	★

[1] significance compared to the mode baseline
[2] significance compared to the random baseline

Table 4: Results of experiments for the facet *topic* (SVM & ext. indexing)

Prob. Indexing & k-NN The use of weights from a learned indexing function for k-NN showed better results than the use of simple binary occurrence weights (see tab. 5). Due to performance reasons and time constraints the sender feature category was ignored and only a single experiment with different features than set of words was performed (as for all other facets). The best result is slightly better than the best result of the ext. indexing with SVM. All results are also statistically significant in respect of both baselines.

experiment	**costs**	**SM**	**SR**
binary weights	0.5562	★	★
binary weights & term mod.	0.5221	★	★
weights by ind. func.	**0.3909**	★	★

Table 5: Results of experiments for the facet *topic* (k-NN & prob. indexing)

Facet *Support Type*

Ext. Indexing & SVM The best result for SVMs were delivered by the term position features (0.7556). Table 6 shows all results. Term features with term modification, tf weights or bigrams worsened the accuracy in comparison to simple binary occurrence weights. Due to the skew class distribution we applied an equal weighting technique to avoid useless results (see [Beckers, 2008] for more details). Attachment features and time features of the non-term features had not proven as useful whereas the other non-term features (punctuation, sender, length, characters) are of value for classification. Most results are statistically significant while a few are only weak or not statistically significant compared to the baselines. In contrast to the facet *topic* not binary term features but term position features have achieved the best result. This supports our hypothesis that also other features should be taken into account for non-topical facets.

experiment	acc.	SM	SR
terms (binary)	0.7393		★
terms (binary & mod.)	0.7240		★
terms (tf)	0.7321	☆	★
terms (bigrams)	0.7199		★
terms	0.7403	☆	★
term position	**0.7556**	★	★
all non-term features	0.2904	★	☆
without punctuation	0.2655	★	
without attachment	0.3065	★	★
without sender	0.2712	★	
without length	0.2730	★	
without time	0.3002	★	★
without characters	0.2774	★	☆
all	**0.7556**	★	★

Table 6: Results of experiments for the facet *support type* (SVM & ext. indexing)

Prob. Indexing & k-NN The usage of weights by a learned indexing function achieved the best results for k-NN (see tab. 7). Term modification also increased the accuracy. Again, all results are statistically significant. Overall, the best result is slightly worse than the best result of the ext. indexing & SVM (0.7271 vs. 0.7556).

experiment	acc.	SM	SR
binary weights	0.72398		★
binary weights & term mod.	0.72403		★
weights by ind. func.	**0.7271**		★

Table 7: Results of experiments for the facet *support type* (k-NN & prob. indexing)

Facet *Priority*

Ext. Indexing & SVM The results of the experiments with SVMs are shown in table 8. The best result with term features only was achieved by terms with binary occurrence weights and term statistics features. As seen before for the other facets, tf weights and bigrams could not increase the classification quality. All non-term features except character features improved the classification quality. The usage of all available features resulted in the lowest costs (0.3967). Most results are statistically significant. Non-term features were able to increase the classification quality together with term features.

Prob. Indexing & k-NN The best accuracy was again achieved with term weights by a learned indexing function. Even the best result of ext. indexing with SVM is outperformed. Term modification was also useful. All results are statistically significant.

5.5 Discussion

Results that are statistically significant better than the baselines can be achieved for all of the three facets. In the following, some other observations we made are described.

experiment	costs	SR
terms (binary)	0.4033	★
terms (binary & mod.)	0.44	★
terms (tf)	0.49	★
terms (special terms)	0.6634	
terms (bigrams)	0.5567	★
terms	**0.3967**	★
term position	0.4167	★
all non-term features	0.4567	★
without punctuation	0.48	★
without attachment	0.56	★
without sender	0.4933	★
without length	0.49	★
without time	0.5067	★
without characters	0.4567	★
all	**0.3833**	★

Table 8: Results of experiments for the facet *priority* (SVM & ext. indexing)

experiment	costs	SR
binary weights	0.4003	★
binary weights & term mod.	0.3475	★
weights by ind. func.	**0.2997**	★

Table 9: Results of experiments for the facet *priority* (k-NN & prob. indexing)

- The estimation of the parameters for SVMs is a very time-consuming task. Some experiments ran several days; in particular, the *topic* facet with 31 classes. Due to the one vs. one classification approach $\frac{k\cdot(k-1)}{2} = \frac{31\cdot(31-1)}{2} = 465$ classifiers had to be learned for a single classification model. The learning of an indexing function with logistic regression took also some days.
- The best results for the facets were achieved by different sets of features. We have shown that it is reasonable to regard also other types of features than just simple set of words. This is in particular the case if classification is performed with respect to a non-topical facet. For the facet *topic* classic sets of words have been the best features.
- Bigrams and tf weights were not useful in any facet. This can be explained due to the fact that bigrams increase the dimensionality of the data. Thus, many bigrams only appear once or twice in the whole collection. Our experiments support that tf as weighting schema has proved to be important for information retrieval but for text classification no such statement can be done.
- Both extended indexing & SVMs and probabilistic, decision-oriented indexing & k-NN have produced results which are statistically significant better than the corresponding baselines. The differences between both techniques were higher for non-topical facets than for the topical facet.

6 Conclusion and Outlook

In comparison to other classification problems it is more difficult to achieve good classification results for the given task. For one thing the quality of the existing classification is rather poor for some facets, especially *priority*. For another thing the difference between two arbitrary classes is not as distinct as e. g. between *spam* and *no spam* in spam classification. Nonetheless, for all facets statistically significant results above the baselines have been achieved.

Extended indexing & SVMs as well as prob. indexing & k-NN have both shown good results. Thus, no conclusion about what technique is generally better for our task can be drawn.

Additional facets, such as e. g. *speech act* or *sentiment*, can be considered. However, our collection does not contain data that is required for these facets. Frommholz and Fuhr [2006] outline some more possible facets. The increasing spreading of d.velop products in other countries than Germany poses new challenges concerning multilingual tickets. The language of a ticket could also be meaningful for classification.

Further improvements could be made with learning techniques that take the classification costs into account during the learning phase (*cost based learning*). Furthermore, feature selection and weighting could increase the classification quality as well as the (time) performance. A more comprehensive evaluation should not only take the multi-facet classification in an isolated way into account but should also investigate whether the multi-facet classification is actually meaningful for employees of the helpdesk and supports them in their daily work.

References

[Abecker *et al.*, 2000] Andreas Abecker, Ansgar Bernardi, Knut Hinkelmann, Otto Kühn, and Michael Sintek. Context-aware, proactive delivery of task-specific knowledge: The KnowMore project. *International Journal on Information System Frontiers (ISF)*, 2((3/4)):139–162, 2000.

[Antoniol *et al.*, 2008] Giuliano Antoniol, Kamel Ayari, Massimiliano Di Penta, Foutse Khomh, and Yann-Gaël Guéhéneuc. Is it a bug or an enhancement?: a text-based approach to classify change requests. Proceedings of CASCON 2008, pages 304–318, New York, NY, USA, 2008. ACM.

[Beckers, 2008] Thomas Beckers. Multifacettenklassifikation von E-Mails im Helpdesk-Szenario. Diploma thesis, Universität Duisburg-Essen, 2008. In German.

[Bekkerman *et al.*, 2004] Ron Bekkerman, Andrew McCallum, and Gary Huang. Automatic categorization of email into folders: Benchmark experiments on Enron and SRI corpora. Technical Report IR-418, University of Massachusetts, CIIR, 2004.

[Bennett and Carbonell, 2005] Paul N. Bennett and Jaime Carbonell. Detecting action-items in e-mail. In *Proceedings of SIGIR 2005*, pages 585–586, Salvador, Brasilien, August 2005. ACM.

[Blanzieri and Bryl, 2006] Enrico Blanzieri and Anton Bryl. A survey of anti-spam techniques. Technical Report DIT-06-056, University of Trento, September 2006.

[Brutlag and Meek, 2000] Jake D. Brutlag and Christopher Meek. Challenges of the email domain for text classification. In Pat Langley, editor, *Proceedings of ICML 2000*, pages 103–110. Morgen Kaufmann, 2000.

[Cohen *et al.*, 2004] William W. Cohen, Vitor R. Carvalho, and Tom M. Mitchell. Learning to classify email into speech acts. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 309–316, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[Cormack, 2007] Gordon V. Cormack. Email spam filtering: A systematic review. *Found. Trends Inf. Retr.*, 1(4):335–455, 2007.

[Crawford *et al.*, 2004] Elisabeth Crawford, Irena Koprinska, and Jon Patrick. Phrases and feature selection in e-mail classification. In *Proceedings of the 9th Australasian Document Computing Symposium*, Melbourne, Australia, December 2004.

[Eichler, 2005] Kathrin Eichler. Automatic classification of swedish email messages. Bachelor thesis, Eberhard-Karls-Universität Tübingen, 2005.

[Frommholz and Fuhr, 2006] Ingo Frommholz and Norbert Fuhr. KI-Methoden zur Email-Archivierung – Technologische Studie zum Inbox-Szenario. Internal Report, Universität Duisburg-Essen, November 2006. `http://www.is.inf.uni-due.de/bib/pdf/ir/Frommholz_Fuhr:06b.pdf`. In German.

[Fuhr and Buckley, 1991] Norbert Fuhr and Chris Buckley. A probabilistic learning approach for document indexing. *ACM Transactions on Information Systems*, 9(3):223–248, 1991.

[Gövert *et al.*, 1999] Norbert Gövert, Mounia Lalmas, and Norbert Fuhr. A probabilistic description-oriented approach for categorising web documents. In Susan Gauch and Il-Yeol Soong, editors, *Proceedings of the Eighth International Conference on Information and Knowledge Management*, pages 475–482. ACM, 1999.

[Hsu and Lin, 2002] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, March 2002.

[Joachims, 1998] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. Forschungsbericht LS-8 Report 23, Universität Dortmund, 1998.

[Klimt and Yang, 2004] Bryan Klimt and Yiming Yang. The Enron corpus: A new dataset for email classification research. In J. G. Carbonell and J. Siekmann, editors, *Proc. of ECML 2004*, volume 3201/2004 of *Lecture Notes in A. I.*, pages 217–226, Pisa, Italy, September 2004. Springer.

[Koprinska *et al.*, 2007] Irena Koprinska, Josiah Poon, James Clark, and Jason Chan. Learning to classify e-mail. *Information Sciences*, 177(10):2167–2187, May 2007.

[Nenkova and Bagga, 2003] Ani Nenkova and Amit Bagga. Email classification for contact centers. In *SAC '03: Proceedings of the 2003 ACM Symposium on Applied Computing*, pages 789–792, New York, NY, USA, 2003. ACM Press.

[Rennie, 2000] Jason D. M. Rennie. ifile: An application of machine learning to email filtering. In Marko Grobelnik, Dunja Mladenic, and Natasa Milic-Frayling, editors, *Proc. of the KDD-2000 Workshop on Text Mining*, Boston, USA, August 2000.

[Sebastiani, 2002] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[Segal and Kephart, 1999] Richard B. Segal and Jeffrey O. Kephart. Mailcat: an intelligent assistant for organizing e-mail. In *AGENTS '99: Proceedings of the third annual conference on Autonomous Agents*, pages 276–282, New York, NY, USA, 1999. ACM Press.

A Tables of features

Feature group	#Features	Description
TERMS	∼	All terms of a ticket
SPECIAL_TERMS	∼	All special terms with a postfix
SPECIAL_TERMS_COUNT	1	number of special terms
BIGRAMS	∼	Bigrams from body and subject
DIFFERENT_TERMS	2	Number of different terms and their relative number in comp. to all terms
TERMS_COUNT	3	Number of terms everywhere, number of terms in body and subject
NO_CHARACTER_OR_DIGIT	2	Number and relative number of special characters
TERMS_IN_FIRST_THIRD	∼	All terms in first third of the body with postfix
TERMS_IN_SECOND_THIRD	∼	All terms in second third of the body with postfix
TERMS_IN_THIRD_THIRD	∼	All terms in third third of the body with postfix
TERMS_IN_SUBJECT	∼	All terms in subject with postfix
TERMS_IN_ATTACHMENT	∼	All terms in attachment with postfix
TERMS_IN_QUESTIONS	∼	All terms in questions with postfix
QUESTION_MARKS	2	Number and relative number of question marks
THREE_QUESTION_MARKS	1	If *???* occurs
EXCLAMATION_MARKS	2	Number and relative number of exclamation marks
THREE_EXCLAMATION_MARKS	1	If *!!!* occurs
HAS_ATTACHMENT	1	If there's an attachment
ATTACHMENT_TYPE	7	type of the attachment (log, text, xml, tmp, image, archive or misc.)
FROM	∼	The sender of the ticket
FROM_COMPANY	∼	The domain part of the sender
FROM_HISTORY	∼	The last few classes of tickets from the sender
FROM_COMPANY_HISTORY	∼	The last few classes of tickets from the sender (domain part)
SUBJECT_LENGTH	1	Number of characters in subject
BODY_LENGTH	1	Number of characters in body
DAY_OF_WEEK	7	The weekday of the ticket
WORKING_HOURS	1	if the time of the ticket is during usual working times
TIME_BLOCKS	∼	Time block of the time of the ticket

Table 10: Table of features (ext. indexing)

Feature group	#Features	Description
termFrequency	1	Frequency of a term in a ticket
inverseDocumentFrequency	1	*idf* of a term
mostFrequentTerms	1	If a term belongs to the most common terms
...	...	...
termInSubject	1	If a terms occurs in the subject
termInAttachment	1	If a term occurs in the attachment
termPositionInBody	3	Three features to indicate where (three thirds) a term appears
termInQuestion	1	If a term occurs in a question

Table 11: Table of features (prob. indexing)

Note: ∼ denotes a variable number of features, because the concrete number depends on the number of terms in a ticket or on other properties of a ticket.

Towards a Geometrical Model for Polyrepresentation of Information Objects

Ingo Frommholz and C. J. van Rijsbergen
Department of Computing Science
University of Glasgow
{ingo|keith}@dcs.gla.ac.uk

Abstract

The principle of polyrepresentation is one of the fundamental recent developments in the field of interactive retrieval. An open problem is how to define a framework which unifies different aspects of polyrepresentation and allows for their application in several ways. Such a framework can be of geometrical nature and it may embrace concepts known from quantum theory. In this short paper, we discuss by giving examples how this framework can look like, with a focus on information objects. We further show how it can be exploited to find a cognitive overlap of different representations on the one hand, and to combine different representations by means of knowledge augmentation on the other hand. We discuss the potential that lies within a geometrical framework and motivate its further development.

1 Introduction

One of the promising recent developments in information retrieval (IR) is the idea of *polyrepresentation*, which came up as a consequence of cognitive theory for interactive IR [Ingwersen and Järvelin, 2005]. The basic idea is that entities may be interpreted or represented in different functional and cognitive ways. Finding relevant documents goes along with finding the *cognitive overlap* of functionally or cognitively different information structures.

We can regard polyrepresentation w.r.t. *information objects* [Skov *et al.*, 2006]. For instance, a Web document can be represented by its content (which reflects the authors view on the document). Nowadays, it is common that users annotate a document in several ways. Annotations may be, for instance, comments, opinions, tags or ratings. Such annotations provide a cognitively different representation of a document, in this case reflecting the users' view on it. Another form of polyrepresentation considers the user's *cognitive state* [Kelly *et al.*, 2005] and different search engines [Larsen *et al.*, 2009]. The former one includes the work task, the perceived information need, the experience and the domain knowledge, and others. The latter one sees different *search engines* as different reflections of the cognitive view of its designers on the retrieval problem. One of the conclusions from evaluating all these facets of polyrepresentation is that the more positive evidence is coming from different representations, the more likely is the object in the cognitive overlap relevant to a given information need.

The experiments on polyrepresentation suggest that search effectiveness can benefit from a retrieval model which explicitly supports the principle of polyrepresentation. What is missing so far is a unified view which incorporates the different facets of polyrepresentation which allows for determining cognitive overlaps, but can go even beyond. For instance, different representations may be combined, as it is possible with knowledge augmentation (see below), to create a new representation. A unified view for polyrepresentation should also consider the combination of the concept of polyrepresentation with the dynamics arising from interactive retrieval. Such a view can be based on a geometrical model, as it was discussed in [van Rijsbergen, 2004]. A growing number of geometrical models, inspired by quantum theory, were introduced recently. For example, Piwowarski and Lalmas propose a geometrical framework which takes into account the evolution of the user's information need (represented as a vector in a Hilbert space) [Piwowarski and Lalmas, 2009]. So far, the concept of polyrepresentation has not been discussed in this model.

The considerations presented here are a first attempt to describe the notion of polyrepresentation (in particular w.r.t. information objects) in a geometrical way. They are a starting point for further discussion into how we can combine the idea of polyrepresentation and geometrical models, possibly inspired by quantum theory.

2 Towards a Geometrical Model

Our discussion starts with an example of how document features in different representations can be expressed geometrically. A representation of a document can be based on a set of distinct features. Such features can be topical, like the appearance of a term in a document, or non-topical, for example the document genre or the page rank. Documents may have static features (like terms and their weights), but they can also be dynamic (e.g., a property which shows whether a document was presented to the user or not). In general, we assume that for a feature f we can estimate the probability $\Pr(f|d)$ that we observe the feature given that we observed d. Similarly, $\Pr(\overline{f}|d) = 1 - \Pr(f|d)$ denotes the probability that we do not observe the feature.

Before we continue our considerations by giving an example, we introduce the notation, which is used in quantum mechanics as well.

2.1 Notation

We give a short introduction to the *Dirac notation*, which we are going to use in the following. A vector x in a real[1] n-dimensional Hilbert space $\mathcal{H}$ can be written as a so-called

[1] Our considerations can be expanded to complex Hilbert spaces, but for the time being it is sufficient to assume that $\mathcal{H}$ is spanned over $\mathbb{R}$

ket in Dirac notation:

$$|\,x\,\rangle = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

with $x_1, \dots x_n \in \mathbb{R}$. Transposed vectors are represented as a *bra*, that is $\langle\, x\,| = (x_1, \dots, x_n)$. Based on this we can define an *inner product* between two vectors x and y as $\langle\, x \mid y\,\rangle = \sum_{i=1}^{n} y_i x_i$ if we assume a canonical basis.

Besides inner products, we can also define an *outer product* as $|\,x\,\rangle\,\langle\, y\,| = xy^T$, which yields a square $n \times n$ matrix in our case. Each such matrix can be regarded as a linear transformation or *operator*. *Projectors* are idempotent, self-adjoint linear operators; they can be used to project vectors onto subspaces. For example, let $|\,e_0\,\rangle = (1,0)^T$ and $|\,e_1\,\rangle = (0,1)^T$ be the base vectors of a two-dimensional vector space $\mathcal{H}$, and $|\,x\,\rangle = (x_1, x_2)^T$ a vector in $\mathcal{H}$. Then $\mathbf{P} = |\,e_0\,\rangle\,\langle\, e_0\,|$ is a projector onto the one-dimensional subspace spanned by $|\,e_0\,\rangle$; $\mathbf{P}\,|\,x\,\rangle = (x_1, 0)^T$ is the projection of $|\,x\,\rangle$ onto that subspace. $||x|| = \sqrt{\sum_i^n x_i^2}$ denotes the *norm* of a vector, and $||x|| = 1$ means the vector is a *unit vector*. If $|\,e_0\,\rangle, \dots, |\,e_n\,\rangle$ form an orthonormal basis of a Hilbert space, then $\mathrm{tr}(\mathbf{T}) = \sum_{i=1}^{n} \langle\, e_i\,|\,\mathbf{T}\,|\,e_i\,\rangle$ is called the *trace* of the matrix $\mathbf{T}$. It is the sum of the diagonal elements of $\mathbf{T}$.

2.2 Polyrepresentation of Information Objects

Representing Document Features

Our basic idea is to encode every feature in a *qubit* (quantum bit). A qubit is a two-dimensional subspace whose base represents two possible disjoint states $|\,0\,\rangle = (1,0)^T$ and $|\,1\,\rangle = (0,1)^T$. We give some examples of how a document feature, in particular a term, can be expressed as a qubit.

Let us assume we have a probabilistic indexer which assigns two probabilities to each term w.r.t. its corresponding document: $\Pr(t|d)$ is the probability that document d could be indexed with t, and $\Pr(\bar{t}|d) = 1 - \Pr(t|d)$ is the probability that it could not. Let $|\,0_t\,\rangle$ and $|\,1_t\,\rangle$ be the base vectors of the qubit for term t. If we set $\alpha = \sqrt{\Pr(t|d)}$ and $\beta = \sqrt{\Pr(\bar{t}|d)}$, then $|\,d\,\rangle = \alpha \cdot |\,1_t\,\rangle + \beta \cdot |\,0_t\,\rangle$ is a unit vector (length 1). The situation is depicted in Figure 1 with $\Pr(t|d) = 0.8$ and $\Pr(\bar{t}|d) = 0.2$ resulting in $|\,d\,\rangle = (d_1, d_2)^T = (\sqrt{0.8}, \sqrt{0.2})^T$ in this qubit.

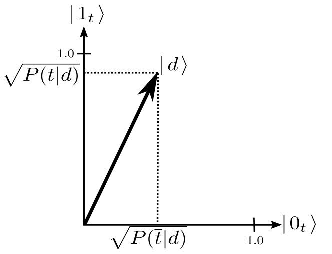


Figure 1: A term feature of d in a qubit

Retrieval with Polyrepresentation Example

Let us assume that we have a collection consisting of two terms, t_1 and t_2, and a document d with a user comment (annotation) a attached to it, so we have two cognitively different representations of the same document. We denote these two representations by two vectors, $|\,d_c\,\rangle$ for the content view and $|\,d_a\,\rangle$ for the annotation view on d. We give an example of how we can derive a simple well-known retrieval function from our representation, namely the traditional vector space model (VSM) which measures the similarity between a document and query vector in a term space. In order to support this, we need to transform our representation based on qubits into the classical vector space representation where the terms are the base vectors. One way to achieve this is to create a new vector $|\,d'_c\,\rangle = (d'_1, d'_2)^T$ with $d'_1 = |\,1_{t_1}\,\rangle\,\langle\, 1_{t_1}\,|\,|\,d_c\,\rangle$ (the projection of $|\,d_c\,\rangle$ onto $|\,1_{t_1}\,\rangle$) and $d'_2 = |\,1_{t_2}\,\rangle\,\langle\, 1_{t_2}\,|\,|\,d_c\,\rangle$. $|\,d'_c\,\rangle$ is then a vector in the classical term space known from the VSM. We can create $|\,d'_a\,\rangle$ out of $|\,d_a\,\rangle$ analogously. The new situation is depicted in Fig. 2. We can see that in contrast to the classical VSM, where the document is represented by only one vector, we now have two vectors for d, namely $|\,d'_c\,\rangle$ and $|\,d'_a\,\rangle$. We further assume that $\sum_{i=1}^{2} \Pr(t_i|d) = 1 = \sum_{i=1}^{2} \Pr(t_i|a)$, which means that $|\,d'_c\,\rangle$ and $|\,d'_a\,\rangle$ are unit vectors. We can represent a content query as a normalised vector $|\,q\,\rangle$ in the term space. We measure the similarity between the query and the two document representations by applying the trace function: $\mathrm{tr}(|\,q\,\rangle\,\langle\, q\,|\,|\,d_c\,\rangle\,\langle\, d_c\,|) = |\,\langle\, q \mid d_c\,\rangle\,|^2$. This equals $\cos^2 \alpha$ (where α is the angle between $|\,d_c\,\rangle$ and $|\,q\,\rangle$) because we assume all vectors to be normalised. The beauty of this is that the resulting similarity value can be interpreted as a probability; see [van Rijsbergen, 2004, p. 83] for further details. By calculating the similarity between $|\,q\,\rangle$ and $|\,d_a\,\rangle$ analogously, we get two probabilities, one for the content representation and one for the representation of the document by its comment. These probabilities can now be used to determine the cognitive overlap of these representations and they can be combined to calculate the final score of d w.r.t. q.

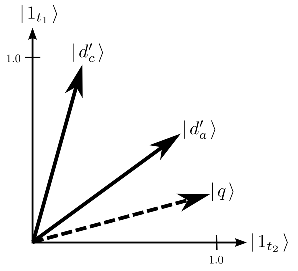


Figure 2: Document representation and query

We have seen that we can derive a well-known retrieval function from our representation. But what is the benefit from expressing features as qubits, when at least for term features we could have created a term space without introducing them? To answer this, we will now give an example of the possible combination of different representations, which relies on the proposed description of features as qubits.

Combination of Representations and Knowledge Augmentation

One may wonder whether the representation of features as a qubit is too redundant, since at least for the term features we also store $\Pr(\bar{t}|d)$, the probability that a document cannot be indexed with a certain term. While in general for other features it might be useful to store this probability as well, it can be incorporated in a senseful way when we want to combine different representations to create a new one. This happens when for example we apply the concept of *knowledge augmentation*. Here, we augment our knowledge about an object with other objects which are connected to it, according to the probability that we actually consider them (see, e.g., [Frommholz and Fuhr, 2006] for a discussion of knowledge augmentation with annotations). Knowledge augmentation basically means to propagate features and their weights from connected objects to the one under consideration. A simple example shall illustrate knowledge augmentation in a geometrical framework. Again, we have a document d and an associated annotation a. We want to augment d with a, which means to propagate all term probabilities in a and also d to an augmented representation of d, denoted d^*. In d^*, the terms (features) and probabilities of d and a are aggregated. Along with a goes $\Pr(c_d^a)$, the probability that we consider a when processing d. One can think of this as a propagation factor[2]. We can store this probability in a qubit as discussed above; the corresponding vector is $|c\rangle = (c_1, c_2)^T = \sqrt{\Pr(c_d^a)} \cdot |1\rangle + \sqrt{1 - \Pr(c_d^a)} \cdot |0\rangle$. Based on this, we can now propagate a term t from d and a to d^* as follows. Qubits can be combined by means of tensor products, and we perform knowledge augmentation by calculating the tensor product of $|\,d\,\rangle$, $|\,c\,\rangle$ and $|\,a\,\rangle$:

$$|\,d^*\,\rangle = |\,d\,\rangle \otimes |\,c\,\rangle \otimes |\,a\,\rangle = \begin{pmatrix} d_1 \cdot c_1 \cdot a_1 \\ d_1 \cdot c_1 \cdot a_2 \\ d_1 \cdot c_2 \cdot a_1 \\ d_1 \cdot c_2 \cdot a_2 \\ d_2 \cdot c_1 \cdot a_1 \\ d_2 \cdot c_1 \cdot a_2 \\ d_2 \cdot c_2 \cdot a_1 \\ d_2 \cdot c_2 \cdot a_2 \end{pmatrix}$$

$|\,d^*\,\rangle$, which represents d^*, is a vector in an 8-dimensional space. The first element of $|\,d^*\,\rangle$ expresses the event that we index d with t (d_1) *and* consider a (c_1) *and* a is indexed with t (a_1). The fifth element denotes the case that we do *not* index d with t (d_2) *and* consider a (c_1) *and* a is indexed with t (a_1). Similarly for the other 6 elements. Each base vector thus represents a possible event, and all these events are disjoint. In fact, the resulting vector represents a probability distribution over these events and is thus a unit vector.

How can we now calculate the probability $\Pr(t|d^*)$ that we observe t in the augmented representation d^*? We observe t in the augmented representation in the following five cases: when we observe it in d, and when we do not observe it in d, but consider a and observe t there. These are exactly the events described by the first 5 elements of $|\,d^*\,\rangle$. These elements contribute to $\Pr(t|d^*)$, whereas the last 3 elements of $|\,d^*\,\rangle$ determine $\Pr(\bar{t}|d^*)$. To get $\Pr(t|d^*)$, we project $|\,d^*\,\rangle$ to the subspace spanned by the first 5 base vectors, and calculate the trace the projection. If $\mathbf{P_t}$ is such a projector, then $\Pr(t|d^*) = \operatorname{tr}(|\,d^*\,\rangle\,\langle\,d^*\,|\,\mathbf{P_t})$. Similarly for $\Pr(\bar{t}|d^*)$. Having achieved both probabilities, we can store them in a qubit as discussed above, and repeat the procedure for the other terms. Note that in this example, we combined a term-based representation with another term-based representation, but we are not bound to this. We can also combine topical and non-topical representations of a document in a similar way.

3 Discussion

We have seen examples for polyrepresentation of information objects in a unified geometrical framework. Document features, be it content features expressed as terms, or non-topical ones, can be represented with the help of qubits which encode the probabilities that a certain feature can be observed or not. In this way, we can integrate different representations of documents in one model, calculate their relevance and use this information to compute the cognitive overlap. Different representations of documents may also be combined, as we have seen for knowledge augmentation. This way, we can exploit the polyrepresentation of information objects to obtain a higher-level representation. This simple example can of course not properly define a whole geometrical framework. This paper is not meant to deliver such a definition, but to undertake a first step towards it and to further motivate it. The following discussion shall reveal what we potentially gain when we further specify a geometrical framework which also includes inspirations coming from quantum mechanics.

We showed an example with different representations of information objects. In fact, also a polyrepresentation of search engines is potentially possible within our framework. How different retrieval models (like the generalised vector space model, the binary independent retrieval model or a language modelling approach) can be described geometrically is reported in [Rölleke *et al.*, 2006]. It is in principal possible to transfer these ideas into our framework, although it has yet to be specified which further knowledge (like relevance judgements) needs to be incorporated. Another extension of the model might also introduce polyrepresentation w.r.t the user's cognitive state, which may be represented as a vector similar to information objects.

The framework discussed in this paper may be used to support other models which indirectly apply polyrepresentation. An example is the Lacostir model introduced in [Fuhr *et al.*, 2008]. This model aims at the integration and utilisation of layout, content and structure (and thus polyrepresentation) of documents for interactive retrieval. The core part of the model consists of certain operations and their resulting system states. For instance, a selection operator (basically a query) lets the user choose relevant documents. Once relevant documents are determined, the user can select suitable representations thereof with a projection operator. An organisation operator can be applied by the user to organise the projected representations, for instance in a linear list or graph. With the visualisation operator, the user can choose between possible visualisations of the organised results. During a session, the user can at any time modify these operators. To support this model, an underlying framework must be capable of handling the different states the user and the system can be in as well as the transitions between them. It also needs to deal with the polyrepresentation of information objects. A geometrical framework can potentially handle the different representations and the dynamics in such a system. At least the selection and projection operators might be mapped to geometrical counterparts, whereas the organisation and visualisation operators may benefit from a geometrical representation of system states as vectors.

While we used some notations borrowed from quantum mechanics, the examples so far are purely classical, but with a geometrical interpretation. They give us a clue of the tight relation between geometry and probability theory and show the potential to embrace existing models in one uni-

[2] A discussion of this probability is beyond the focus of this paper. It might be system-oriented, e.g. determined by the number of comments, or user-oriented, for instance by rating comments as important or less important.

fied framework. However, we did not touch any concepts used in quantum mechanics yet, like entanglement or complex numbers. For instance, different representations of an information object can be related, a property which we apply with knowledge augmentation. This relationship may also be expressed by using one state vector per feature and document, but with a different basis for each representation. Different representations may be entangled, and such property could easily be included in our model. An open question therefore is how the relationship between different representations should be modelled.

4 Related Work

The idea of using geometry for information retrieval, going far beyond the VSM, was formulated in [van Rijsbergen, 2004]. In this book, the strong connection between geometry, probability theory and logics is expressed. The examples in this paper are further inspired by Melucci's work reported in [Melucci, 2008]. Here, contextual factors (which may be different representations of information objects, but also reflect the user's point of view) are expressed as subspaces. Information objects and also queries are represented as vectors within these subspaces. Given this representation, a probability of context can be computed. This resembles the idea sketched in this paper, but the approach is not focused on polyrepresentation of objects. In the model presented by Piwowarski and Lalmas, a user's information need is represented as a state vector in a vector space which may for instance be set up by (possibly structured) documents [Piwowarski and Lalmas, 2009]. Initially, less is known about the actual information needs. Each user interaction gains more knowledge about her information need, which lets the state vector collapse until the information need is expressed unambiguously. Schmitt proposes QQL, a query language which integrates databases and IR [Schmitt, 2008]. In his work, he makes use of qubits as the atomic unit of retrieval values and interrelates quantum logic and quantum mechanics with database query processing. Further approaches about the relation of quantum theory and IR are reported in the proceedings of the Quantum Interaction symposium (see, e.g., [Bruza *et al.*, 2009]).

5 Conclusion and Future Work

In this short paper, we showed by an example how polyrepresentation of information objects can be realised geometrically. The goal is to undertake a first step towards a unified framework for polyrepresentation, which is missing so far. The example also shows how we can geometrically combine different representations to a new one. A subsequent discussion reveals some of the further possibilities coming from a geometrical approach.

We will also investigate the integration of existing quantum-inspired models into the framework, like the ones reported in [Piwowarski and Lalmas, 2009] or [Melucci, 2008], which do not deal with polyrepresentation yet. These models may thus be extended with the ideas that came up in the discussion so far, like knowledge augmentation and the possible entanglement of representations.

6 Acknowledgements

Our work was supported by the EPSRC project *Renaissance*[3] (grant number EP/F014384/1).

[3] http://renaissance.dcs.gla.ac.uk/

References

[Bruza *et al.*, 2009] Peter Bruza, Donald Sofge, William Lawless, Keith van Rijsbergen, and Matthias Klusch, editors. *Proceedings of the Third International Symposium on Quantum Interaction (QI 2009)*, LNCS, Heidelberg et al., March 2009. Springer.

[Frommholz and Fuhr, 2006] Ingo Frommholz and Norbert Fuhr. Probabilistic, object-oriented logics for annotation-based retrieval in digital libraries. In M. Nelson, C. Marshall, and G. Marchionini, editors, *Proc. of the JCDL 2006*, pages 55–64, New York, 2006. ACM.

[Fuhr *et al.*, 2008] Norbert Fuhr, Matthias Jordan, and Ingo Frommholz. Combining cognitive and system-oriented approaches for designing IR user interfaces. In *Proceedings of the 2nd International Workshop on Adaptive Information Retrieval (AIR 2008)*, October 2008.

[Ingwersen and Järvelin, 2005] P. Ingwersen and K. Järvelin. *The turn: integration of information seeking and retrieval in context*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[Kelly *et al.*, 2005] Diane Kelly, Vijay Deepak Dollu, and Xin Fu. The loquacious user: a document-independent source of terms for query expansion. In *Proceedings of the SIGIR 2005*, pages 457–464, New York, NY, USA, 2005. ACM.

[Larsen *et al.*, 2009] Birger Larsen, Peter Ingwersen, and Berit Lund. Data fusion according to the principle of polyrepresentation. *Journal of the American Society for Information Science and Technology*, 60(4):646–654, 2009.

[Melucci, 2008] Massimo Melucci. A basis for information retrieval in context. *Information Processing & Management*, 26(3), June 2008.

[Piwowarski and Lalmas, 2009] Benjamin Piwowarski and Mounia Lalmas. Structured information retrieval and quantum theory. In Bruza et al. [2009], pages 289–298.

[Rölleke *et al.*, 2006] Thomas Rölleke, Theodora Tsikrika, and Gabriella Kazai. A general matrix framework for modelling information retrieval. *Information Processing and Management*, 42(1):4–30, 2006.

[Schmitt, 2008] Ingo Schmitt. QQL: A DB&IR query language. *The International Journal on Very Large Data Bases*, 17(1):39–56, 2008.

[Skov *et al.*, 2006] Mette Skov, Birger Larsen, and Peter Ingwersen. Inter and intra-document contexts applied in polyrepresentation. In *Proceedings of IIiX 2006*, pages 97–101, New York, NY, USA, 2006. ACM.

[van Rijsbergen, 2004] C. J. van Rijsbergen. *The Geometry of Information Retrieval*. Cambridge University Press, New York, NY, USA, 2004.

Methoden für Robustes Information Retrieval und dessen Evaluierung

Thomas Mandl, Daniela Wilczek
Institut für Informationswissenschaft und Sprachtechnologie
Universität Hildesheim
Marienburger Platz 22
31141 Hildesheim, Deutschland
mandl@uni-hildesheim.de

Abstract

Information Retrieval Systeme sollen möglichst robust arbeiten und ihren Benutzern unter einer Vielzahl von Bedingungen zufriedenstellende Ergebnisse liefern. Der Beitrag referiert Ansätze zur Entwicklung von robusten Systemen und zeigt, wie die Robustheit evaluiert werden kann. Die Ergebnisse des Robust Task des Cross Language Evaluation Forum (CLEF) 2008 werden vorgestellt und unter verschiedenen Gesichtspunkten analysiert.

1 Robustheit im Information Retrieval

Wissen muss in vielerlei Kontexten und unter sehr heterogenen Bedingungen aktiviert werden und als Information den Benutzer erreichen. Robustheit ist daher ein Desiderat für Information Retrieval Systeme.

Eine in der Praxis häufig angewandte Form der Evaluierung von Information Retrieval Systemen ist die Ego-Search. Dabei gibt jemand seinen eigenen Namen in das Eingabefeld für die Suche und wirft einen Blick auf die Ergebnisse. Diese Form der Evaluierung hält wissenschaftlichen Kriterien nicht stand. Prüfen wir einige dieser Kriterien ab und vergleichen wir die Ego-Search mit einer wissenschaftlichen Evaluierung nach dem Cranfield-Paradigma. Die Nutzung nur einer Anfrage führt zu einem völlig unzuverlässigem Ergebnis. Typischerweise nutzen Evaluierungen 50 Anfragen und mitteln die Einzelergebnisse. Wird nur ein System betrachtet, so lässt sich überhaupt keine Aussage über Qualität machen. Betrachtet der Juror in der Ego-Search nur die obersten Treffer, so stellt dies ebenfalls keinen gerechten Maßstab dar und schränkt das Ergebnis auf Anwendungsfälle ein, die eine hohe Präzision erfordern. Viele Szenarien erfordern aber einen hohen Recall.

Die Evaluierungsforschung kann seit einigen Jahren auf umfangreiche Daten aus den Initiativen Text Retrieval Conference (TREC), Cross Language Evaluation Forum (CLEF) und NTCIR zurückgreifen. So konnten genaue Analysen der Zuverlässigkeit von Retrieval-Tests erfolgen und viele der früher eher heuristischen Annahmen einer wissenschaftlichen Überprüfung unterziehen.

2 Aktuelle Forschung zur Evaluierung

Die Evaluierungen von Information Retrieval Systemen folgen meist dem Cranfield-Paradigma, welches die Bestandteile einer zuverlässigen vergleichenden Bewertung benennt [Robertson 2008]. Mehrere Systeme bearbeiten identische Aufgaben (Topics) für eine Menge von Dokumenten. Im Anschluss werden die Treffer zusammengefasst und von menschlichen Juroren bewertet. Je nach der Position der relevanten und nicht-relevanten Dokumente in den Trefferlisten der Systeme können Qualitäts-Maße berechnet werden, anhand derer die Systeme verglichen werden.

Die Validität des Cranfield-Paradigmas wurde in den letzten Jahren aus mehreren Blickwinkeln analysiert. Wie viele Experimente, Systeme, Dokumente, Anfragen, Juroren und Relevanzurteile sind eigentlich nötig, um zwei Systeme verlässlich vergleichen zu können und Rückschlüsse auf ihre Performanz außerhalb der Evaluierung ziehen zu können?

Häufig wurde etwa die Subjektivität der Juroren bemängelt. Manche Forscher fanden weitere, aus ihrer Sicht relevant Dokumente unter den bisher nicht bewerteten oder den als negativ bewerteten Dokumenten. Die scheint den Relevanz-Urteilen als Fundament der Bewertung die Glaubwürdigkeit zu entziehen. Anhand von mehrfach bewerteten Dokumenten für TREC 4 nachgewiesen werden, dass tatsächlich bei den relevanten Dokumenten tatsächlich nur eine Übereinstimmung von zwischen 30% und 40% erreicht wurde [Voorhees 1998]. Gleichwohl wirkt sich dies aber nicht auf die Reihenfolge der Systeme aus. Der Vergleich zwischen den Systemen fällt unabhängig von Juroren sehr ähnlich aus [Voorhees 1998].

Die Variabilität zwischen den Topics ist bei allen Evaluierungen meist größer als die zwischen den Systemen. Dies wirft erneut berechtigen Zweifel an der Zuverlässigkeit der Evaluierung auf. Wenn die Topics sich sehr stark voneinander unterscheiden, dann könnte die zufällige Auswahl von einigen anderen Topics doch zu einem stark abweichendem Evaluierungsergebnis führen. Auch dies wurde mehrfach wissenschaftlich untersucht. Dazu geht man vom Original-Ranking der Systeme aus, lässt einzelne Topics weg und betrachtet die Ergebnisse der Evaluierung ohne diese Topics. Das Ergebnis ist eine Rangfolge von Systemen. Unter der zwar fragwürdigen, aber plausiblen Annahme, dass das Ranking mit allen Topics das Optimum darstellt, kann nun das optimale mit dem neuen, mit weniger Topics erstellen Ranking verglichen werden. Korrelationen oder Fehler-Maße, die zählen, wie oft ein „schlechteres“ System vor einem besseren platziert ist, liefern Maßzahlen für den Vergleich.

Hier zeigte sich immer, dass die Rangfolgen der Systeme weitgehend stabil blieben und sich bis zu einer Menge von etwa 25 Topics wenig an dem Ergebnis der Evaluierung ändert [Sanderson & Zobel 2005, Mandl 2008]. Auch hinsichtlich der Anzahl der Topics sind die typischen Evaluierung also sehr zuverlässig. Ab 25 Anfragen kann man mit einem guten Level an Zuverlässigkeit rechnen.

Robustheit bedeutet für Produkte gemeinhin, dass sie auch unter wechselnden, schwierigen und unvorhergesehenen Bedingungen noch einigermaßen zufriedenstellend funktionieren, wobei in Kauf genommen wird, dass sie nicht unbedingt besonders glänzen. Für Information Retrieval Systeme kommen hierfür unterschiedliche Kollektionen oder Benutzer in Frage, jedoch liefert vor allem die bereits besprochene Variabilität zwischen einzelnen Aufgaben die größte Herausforderung.

Die Qualität von Antworten im Information Retrieval schwankt zwischen einzelnen Anfragen sehr stark. Die Evaluierung im Information Retrieval zielt in der Regel auf eine Optimierung der durchschnittlichen Retrieval-Qualität über mehrere Testanfragen (Topics). Sehr schlecht beantwortete Anfragen wirken sich besonders negativ auf die Zufriedenheit des Benutzers aus. Für die Steigerung der Robustheit ist es erforderlich, besonders die Qualität der Systeme für die schwierigen Topics zu erhöhen [Voorhees 2005]. Dazu ist sowohl die automatische Vorhersage der Schwierigkeit als auch die Analyse und besondere Behandlung der vermutlich schwierigen Topics nötig. Dementsprechend verspricht die Analyse der einzelnen Topics großes Potential für die Verbesserung der Retrieval-Ergebnisse [Mandl 2008].

3 Entwicklung Robuster Systeme

Für die Steigerung der Robustheit von Information Retrieval Systemen wurden bisher explizit einige Verfahren getestet. Bereits die Grundformreduktion kann als robuste Technik angesehen werden. Die robusten Techniken dienen meist zur der Steigerung der Performanz bei schwierigen Topics, jedoch können auch andere generelle Strategien zur Klassifizierung von Topics und ihrer spezifischen Behandlung durch bestimmte Systemkomponenten als robuste Techniken gelten.

Schwierige Anfragen entstehen oft dadurch, dass der Suchbegriff in der Kollektion gar nicht enthalten ist (*out of vocabulary* Problem). Dadurch entfalten Strategien zur Erhöhung des Recall wie automatisch generierten Ähnlichkeitsthesauri und Blind Relevance Feedback gar nicht die notwendige Wirkung. Als Strategie hat sich hier die Termerweiterung in einer anderen Kollektion bewährt. Dabei wird der nicht gefundene Begriff z.B. im Web gesucht, aus den Treffern werden häufig mit ihm in Beziehung stehende Terme extrahiert und mit diesen wird dann in der gewünschten Kollektion recherchiert.

Aber auch andere Parameter der Termerweiterung werden analysiert [Tomlinson 2007]. In einem Experiment im Rahmen von CLEF (siehe Abschnitt 5) berichten [Pérez-Agüera & Zaragoza 2008], dass Blind Relevance Feedback (BRF) zwar die durchschnittliche Qualität des Systems steigert (*Mean Average Precision*, MAP), jedoch die Robustheit verringert (*Geometric Mean Average Precision*, GMAP).

Dieses Phänomen hat [Kwok 2005] ebenfalls für das Englische untersucht und dabei die Auswirkungen von Blind Relevance Feedback auf unterschiedlich schwierige Anfragen analysiert. Es zeigte sich, dass BRF für mittelschwere Aufgaben zu positiven Effekte führte, während es bei schwierigen und leichten Anfragen eher Verschlechterungen bewirkte. Dies lässt sich möglicherweise dadurch erklären, dass leichte Anfragen bereits im ersten Durchlauf relative gut beantwortet werden und BRF weniger relevante Terme hinzufügt. Dagegen funktioniert BRF bei sehr schwierigen Anfragen nicht, weil das erste Ergebnis so schlecht ist, dass keine guten Erweiterungsterme gefunden werden [Kwok 2005].

Darauf aufbauende Arbeiten versuchen, die Kohärenz der besten Treffer zu untersuchen. Die Kohärenz in der gesamten Kollektion und den inhaltlichen Zusammenhang der besten Treffer vergleichen [He et al. 2008]. Dabei bemerken sie, dass BRF besser wirkt, wenn die Treffermenge thematisch sehr homogen ist. Treten Dokumente zu mehreren Themen in der Treffern auf, so kann das BRF eine thematische Verschiebung bewirken. Eine Messung der Kohärenz in der Treffermenge kann über die Anwendung des BRF entscheiden.

Auch ambige Terme können bei der Termerweiterung eher schaden als nutzen. In einem System bestimmen [Cronen-Townsend et al. 2002] ein Klarheitsmaß, welches die Ambiguität eines Terms im Vergleich zum *language model* der Kollektion misst. Nur nicht ambige Terme werden zur Termeweiterung eingesetzt.

Ein weiterer Ansatz zur Verbesserung der Robustheit von Systemen liegt in der automatischen Disambiguierung. Ambiguität tritt in der natürlichen Sprache häufig auf und führt zu Schwierigkeiten beim Retrieval. Eine Analyse des Kontexts kann helfen, aus mehreren Bedeutungen eines Wortes die gemeinte zu erschließen. Dieser Ansatz wurde u.a. in CLEF 2008 untersucht.

4 Robuste Evaluierung

Die CLEF-Initiative (www.clef-campaign.org) etablierte sich im Jahr 2000. Seitdem steigt die Zahl der Teilnehmer bei CLEF stetig an und die Forschung zu mehrsprachigen Information Retrieval Systemen gewinnt an Dynamik. CLEF folgt dem Modell von TREC und schuf eine mehrsprachige Kollektion mit Zeitungstexten. Inzwischen umfasst die Dokument-Kollektion für das ad-hoc Retrieval die Sprachen Englisch, Französisch, Spanisch, Italienisch, Deutsch, Holländisch, Schwedisch, Finnisch, Portugiesisch, Bulgarisch, Ungarisch und Russisch. Mehrere weitere Tracks wie *Question Answering*, *Web-Retrieval*, *Spoken Dokument Retrieval* oder *Geographic CLEF* untersuchen bestimmte Aspekte des mehrsprachigen Retrieval.

Im Rahmen von CLEF wird seit drei Jahren ein Robust Task durchgeführt, der benutzerorientierte Maße wie GMAP in den Mittelpunkt rückt [Mandl 2006]. Für CLEF 2008 standen erstmals Disambiguierungs-Daten für die Dokumente zur Verfügung, so dass überprüft werden konnte, ob zusätzliches semantisches Wissen das Retrieval robuster gestalten kann [Agirre et al. 2009].

5 Ergebnisse des Robust Task 2008

Die Ergebnisse der Robust Task 2008 haben gezeigt, dass sich die Hoffnung nicht unmittelbar bestätigt, dass Disambiguierungs-Daten (Word Sense Disambiguation, WSD) das Retrieval verbessern. Im mono-lingualen Retrieval für das Englische berichteten zwar einige Forschungsgruppen, dass sie durch WSD bessere Ergebnisse erzielen konnten. Das beste Ergebnis war aber ohne WSD erzielt worden. Allerdings galt dies nur für MAP. Berücksichtigt man den GMAP, so ergibt sich eine Veränderung auf der ersten Position und das WSD-Experiment der gleichen Gruppe rückt auf den ersten Platz [Agirre et al. 2009].

Im Folgenden werden die Ergebnisse für den bilingualen Task vom Spanischen ins Englisch betrachtet. Hier sieht die Bilanz für WSD noch negativer aus. Abbildung 1 zeigt die Verteilung der *Average Precision* über den Wertebereich. Der Box-Plot zeigt an den Antennen den minimalen und den maximalen Wert auf, während die Box mit dem Strich die mittleren 50% der Werte mit dem Median markiert. Zum einen ist die Spannbreite der Topics und die der Systeme aufgezeichnet. Jeweils ein Box-Plot zeigt die Systeme, die WSD benutzen und die Systeme, die ohne WSD arbeiten. Bei den Topics sind jeweils alle Topics einmal für die Systeme mit und einmal die ohne WSD gezeigt.

Die Abbildung zeigt deutlich, dass wie bei jeder Evaluierung die Varianz bei den Topics sehr viel höher ist als die bei den Systemen. Bei den Topics (Aufgaben) sind immer sowohl sehr schwere als auch sehr leichte Topics vertreten. Das Minimum der Topics liegt immer bei 0 und das Maximum immer über 0,8. Die Spannbreite der Systeme ist sehr viel geringer. An beiden Verteilungen ist jeweils aber zu erkennen, dass WSD anscheinend eher schadet. Der Median und das Maximum liegen bei Benutzung von WSD niedriger. Disambiguierungs-Daten scheinen die Robustheit nicht zu erhöhen.

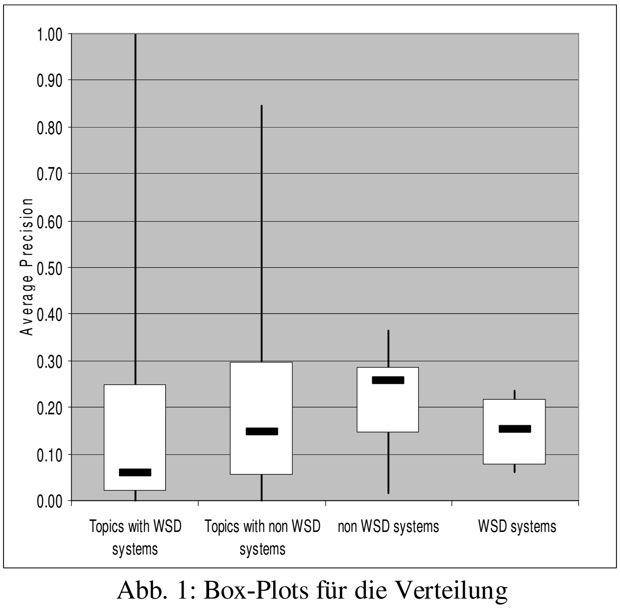


Abb. 1: Box-Plots für die Verteilung der AP für Topics und Systeme

Bei der vergleichenden Betrachtung der MAP und des GMAP ergeben sich besonders unter Berücksichtigung kleinerer Topic-Mengen interessante Effekte. Besonders sollte untersucht werden, ob bestimmte Topics besonders von der WSD profitiert haben und welche Topics unter der WSD gelitten haben. Die folgenden Tabellen listen diese Topics. Diese können für die Analyse der Gründe für das Scheitern oder den Nutzen von WSD gute Hinweise bieten. Diese liefern möglicherweise Hinweise für weitere Verbesserungsansätze.

Insgesamt standen 160 Testfragen zur Verfügung. Daraus wurden mehrere kleinere Mengen erzeugt. Zum einen zwei 50er und eine 60er Gruppe, wobei die ersten 50 eine Gruppe bildeten und die zweiten 50 eine zweite und der Rest die 60er Gruppe. Für die Erzeugung zweier 100er und zweier 120er Gruppen wurde der gleiche Ansatz gewählt, wobei beide Gruppen sich in der Mitte jeweils überlappen. Variiert wurde die Reihenfolge der Topics, so dass mehrere verschiedene Versionen v.a. der 50er Aufteilung erzeugt wurden.

Tabelle 1: Verschlechterung durch WSD (-0,48 bis -0,26 MAP absolut)

170	Find documents about French plans for reducing the number of official languages in the European Union to five languages.
333	Find information on the trial of the French war criminal, Paul Touvier.
185	What happened to the photographs and films that Dutch soldiers made in Srebrenica which provided evidence of violations of human rights?
289	Find documents giving information on the Falkland Islands
162	Find documents about the problems posed by Greece concerning the abolishment of customs restrictions between the European Union and Turkey

Tabelle 2: Verbesserung durch WSD (+0,16 bis +0,26 MAP absolut)

183	In what parts of Asia have dinosaur remains been found?
173	Find reports on the experimental proof of top quarks by US researchers
294	What is the speed of winds in a hurricane?
253	In which countries or states is the death penalty still practiced or at least permitted by the constitution?
196	Find reports on the merger of the Japanese banks Mitsubishi and Bank of Tokyo into the largest bank in the world.

Jede dieser Gruppen wurde als individueller Retrieval-Test betrachtet und daraus eine Rangfolge der Systeme sowohl mit MAP als auch mit GMAP ermittelt. Insgesamt waren die Korrelationen zwischen Rangfolgen sehr hoch. Für die 160 Topics ergab sich eine Korrelation von 0.90. Das bedeutet, dass es minimale Änderungen bei den Positionen gab. Interessanterweise liegen die meisten Korrelationen für kleinere Mengen höher. Geringere Korrelationswerte ergeben sich lediglich, wenn die 50er Untermengen aus einer Sortierung nach der Verbesserung der AP durch die WSD erzeugt werden. Für die beiden letzten Mengen, also die Gruppen der Topics welche durch WSD eher profitieren, liegt die Korrelation nur bei 0,87. Bei diesen 50 bzw. 60 Topics macht es also durchaus Sinn, den GMAP zu betrachten und damit ein robustes Maß anzuwenden.

6 Fazit

Dieser Aufsatz erläutert die robuste Systementwicklung ebenso wie die robuste Evaluierung. Einige Ergebnisse aus der Analyse von Evaluierungsergebnissen werden vorgestellt. Auch im Rahmen von CLEF 2009 findet wieder ein Robust Task statt.

Literatur

Agirre, E.; Di Nunzio, G.; Ferro, N.; Mandl, T.; Peters, C. (2009). CLEF 2008: Ad Hoc Track Overview. In: Evaluating Systems for Multilingual and Multimodal Access: 9th Workshop of the Cross-Language Evaluation Forum, CLEF 2008, Aarhus, Denmark, Revised Selected Papers. Berlin et al.: Springer [LNCS]. erscheint. Preprint: http://www.clef-campaign.org

[Cronen-Townsend et al. 2002] Cronen-Townsend, S.; Zhou, Y.; Croft, B. (2002). Predicting query performance. In: 25th Annual Intl. ACM conference on Research and development in information retrieval (SIGIR). Tampere. S. 299-306.

[He et al. 2008] He, J.; Larson, M.; de Rijke, M. (2008). On the Topical Structure of the Relevance Feedback Set. In: Proc. LWA: Workshop Information Retrieval. http://ki.informatik.uni-wuerzburg.de/papers/baumeister/2008/LWA2008-Proc.pdf

[Kwok 2005] Kwok, K. (2005). An Attempt to Identify Weakest and Strongest Queries. In: SIGIR Workshop Predicting Query Difficulty. Salvador, Brazil. http://www.haifa.il.ibm.com/sigir05-qp

[Mandl 2006] Mandl, T. (2006). Benutzerorientierte Bewertungsmaßstäbe für Information Retrieval Systeme: Der Robust Task bei CLEF 2006. In: Mandl, T.; Womser-Hacker, C. (Hrsg.): Effektive Information Retrieval Verfahren in Theorie und Praxis: Proc. des Fünften Hildesheimer Evaluierungs- und Retrievalworkshop (HIER 2006). Hildesheim: Universitätsbibliothek. S. 79-91. http://web1.bib.uni-hildesheim.de/edocs/2006/ 519937899/meta/

[Mandl 2008] Mandl, T. (2008). Die Reliabilität der Evaluierung von Information Retrieval Systemen am Beispiel von GeoCLEF. In: Datenbank-Spektrum: Zeitschrift für Datenbanktechnologie und Information Retrieval. Heft 24. S. 40-47

[Pérez-Agüera & Zaragoza 2008] Pérez-Agüera, J; Zaragoza, H. (2008). UCM-Y!R at CLEF 2008 Robust and WSD tasks. In: CLEF Working Notes. http://www.clef-campaign.org

[Robertson 2008] Robertson, S. (2008): On the history of evaluation in IR. In: Journal of Information Science. http://jis.sagepub.com/cgi/reprint/34/4/439

[Sanderson & Zobel 2005] Sanderson, M.; Zobel, J. (2005). Information retrieval system evaluation: effort, sensitivity, and reliability. In: 28th Annual Intl. ACM Conference on Research and Development in Information Retrieval (SIGIR) Salvador, Brazil. S. 162-169

[Tomlinson 2007] Tomlinson, S. (2007). Comparing the Robustness of Expansion Techniques and Retrieval Measures. In: Evaluation of Multilingual and Multimodal Information Retrieval. 7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006, Alicante, Spain, Revised Selected Papers [LNCS 4730] S. 129-136.

[Voorhees 1998] Voorhees, E. (1998). Variations in relevance judgments and the measurement of retrieval effectiveness. In: 21st Annual Intl. ACM Conference on Research and Development in Information Retrieval (SIGIR) Melbourne. S. 315-323

[Voorhees 2005] Voorhees, E. (2005). The TREC robust retrieval track. In: ACM SIGIR Forum 39 (1) 11-20.

How Users Search in the German Education Index – Tactics and Strategies

Carola Carstens, Marc Rittberger, Verena Wissel
German Institute for International Educational Research
Information Center for Education, Frankfurt, Germany
carstens@dipf.de, rittberger@dipf.de, w.verena@t-online.de

Abstract

In this empirical study, search tactics and strategies of the users of the German Education Index, a bibliographic database for the domain of educational research, were analyzed. With the help of logfile analyses, the use of 15 different term tactics, 4 search formulation tactics and 6 distinct search strategies could be identified. As a first step towards gaining a deeper understanding of complex subject search processes, the quantitative occurrences of these tactics and strategies are presented in this paper, augmented with illustrative examples for each of them.

1 Motivation

Although much research exists on the search behavior of users of web search engines, these results are not necessarily transferrable to the search in specialized search engines. Consequently, the question arises if the findings for web search behavior also apply to specialized search engines, one of which is the German Education Index[1].

For example, a well-known study by [Spink et al. 2001] states that web searches on average consist of only 2,4 terms, that advanced search features are only scarcely used and that few query reformulations occur. The study at hand analyzed if this also holds true for the German Education Index.

Moreover, the study aimed to get an insight into the users' subject search behavior in general and to understand commonly applied search tactics and strategies. In order to analyze user's reformulation tactics and to examine if they show a tendency for certain search strategies, the study's focus was on complex search sessions. Based on these results, search support functions may be deducted and implemented in the future to optimally assist the users during the query formulation process.

2 Related Work

2.1 Studies on search behavior

To assess the search behavior of users of a web search engine, [Spink *et al.*, 2001] analyzed the logfiles of one day of the Excite search engine. They came to the conclusion that "most people use few search terms, few modified queries, view few Web pages, and rarely use advanced search features". But they also state that this may be characteristic of web searching and that this does not necessarily apply to other retrieval systems. They found out that 48,4% of the users submitted a single query, 20,8% two queries and 31% three or more queries. If queries were reformulated, modifications appeared in small increments, preferably by changing, adding or deleting one term at a time. The average number of terms per query was 2,4 and less than 5% of the queries used Boolean operators. On average, users posted 4,86 queries per search session.

In a previous study on the Excite data set, [Jansen *et al.*, 2000] reported an average of 2,8 queries per search session but also stated that most users use just one query per search. [Lau and Horvitz, 1999] further analyzed the semantics of query refinements in Excite searches and state that few users refined their searches by specialization, generalization or reformulation.

[Rieh and Xie, 2006] analyzed Excite searches not only quantitatively but traced query reformulation sequences with the help of query logs. They focused on 313 sessions with six or more unique queries. In this excerpt, they could distinguish eight reformulation patterns - *specified, generalized, parallel, building-block, dynamic, multitasking, recurrent* and *format reformulation* patterns, for which they present illustrative examples. Furthermore, they analyzed the use and frequency of the following content related reformulation tactics - *specification* (29,1%), *generalization* (15,8%), *replacement with synonym* (3,7%) and *parallel movements* (51,4%).

[Silverstein and Henzinger, 1999] analyzed query logs of the AltaVista Search Engine and also came to the conclusion that users mostly defined short queries of an average length of 2,35 and scarcely use query modifications.

Apart from studies focusing on search behavior in web search engines, several studies exist that refer to domain-specific retrieval systems. They thus have a similar application domain as the study at hand.

For example, [Sutcliffe *et al.*, 2000] conducted an empirical study on the MEDLINE database and assessed the search behavior of information novices and experts. They found out that the average recall of all subjects was low (13,94%), compared to a gold standard. On the whole, novices used fewer query iterations than experts. For example, experts used cycles of narrowing and broadening, whereas novices focused on trial and error approaches. Moreover, experts used facilities like term suggestions, thesaurus and term exploration (truncation

[1] http://www.fachportal-paedagogik.de/fis_bildung/fis_form_e.html

and wild cards) more frequently than novices and also made more use of Boolean operators. On average, experts used 9,1 terms per query, while novices used 6,6 terms.

[Wildemuth and Moore, 1995] also analyzed search behavior on the MEDLINE database, coming to the conclusion that the query formulations could be improved by a more frequent use of synonyms, the correct use of Boolean operators and the more frequent consultation of controlled vocabulary resources such as an online thesaurus.

2.2 Search tactics and strategies

Basic concepts for the description and analysis of search behavior were defined by [Bates, 1979] and [Harter, 1986]. [Bates, 1979] describes search processes in terms of search tactics and search strategies. While the search strategy designates the overall search plan, a tactic is defined as "a move made to further a search", thus serving to realize the superordinate strategy.

According to [Bates, 1979], four types of search tactics can be distinguished, two of which are term tactics and search formulation tactics. She lists the following different search formulation tactics - *specify, exhaust, reduce, parallel, pinpoint and block*, which all describe actions to design or reformulate a query, for example by using Boolean operators.

Term tactics, by contrast, apply specifically to certain terms in the query, which may be added, replaced or deleted, for example by the following tactics - *super, sub, relate, neighbor, trace, vary, fix, rearrange, contrary, respell, respace*, as listed by [Bates, 1979].

A combination of search tactics may be used to pursue a certain search strategy. [Harter, 1986] draws a distinction between subject searches, that focus on the document contents, and non-subject searches, which query for non-semantic characteristics of documents. To achieve the latter goal, non-subject searches can make use of certain query fields such as *document type, year of publication, language, author* or *source*.

Compared to non-subject searches, subject searches show a big variety of complexity, ranging from simple searches following the *quick approach* to very sophisticated approaches such as in the *pairwise facets* strategy.

As defined by [Chu 2003], the *quick approach* is the simplest way of searching. The user enters one or more terms without using any operators.

The *briefsearch* strategy [Harter, 1986] adds one level of complexity to the *quick approach* as Boolean operators are used. It primarily serves to get an overview of the documents in the retrieval system. For this reason, it is often used as an entry point to a more complex search. If several retrieval systems are queried at the same time, [Harter, 1986] speaks of a *multiple briefsearch* strategy.

For example, a *briefsearch* can serve as an entry point to the *building blocks* approach. In this strategy, the search is split up into equivalent facets, each of which can be represented by several terms. While the terms in each facet are connected by the Boolean OR operator, the distinct facets may be combined by both OR or AND operators.

[Harter, 1986] further enumerates different kinds of successive facet strategies. Depending on the nature of the search facet that is used in the first query step, he distinguishes the *most specific concept first* strategy and the *fewest postings first* strategy.

If the search facets are considered to be equally important, the *pairwise facets* strategy [Harter, 1986] can be employed. In this case, two facets at a time are intersected, and finally the result sets of all facet combinations are merged.

While the above mentioned strategies start with a high recall, the *citation pearl growing approach* [Harter, 1986] starts with a precision-oriented search to identify a few relevant documents from which new search terms can be inferred.

In *interactive scanning*, the user starts with a high-recall search and scans the possibly long result lists, which is a very time-expensive approach.

If a domain-specific search system is used, facets that represent the domain focus should not be used. [Harter, 1986] calls this strategy *implied facets*.

Other strategies that take the result documents' citations into account are subsumed as *citation indexing strategies* by [Harter, 1986]. They are based on the assumption that cited publications, authors or cocited authors lead to relevant documents.

3 The German Education Index

The German Education Index is a bibliographic database for the domain of educational research. In April 2009, it comprised 657720 documents, primarily in German language (more than 80%). The corpus can be searched by a retrieval system that is based on the Lucene search engine.

The quick search mode (figure 1) allows to dispatch free text queries which apply to the document fields *index terms, title, author/editor, institutions, abstract* and *source*.

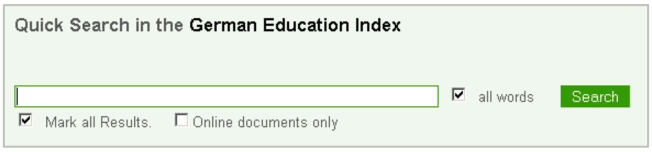
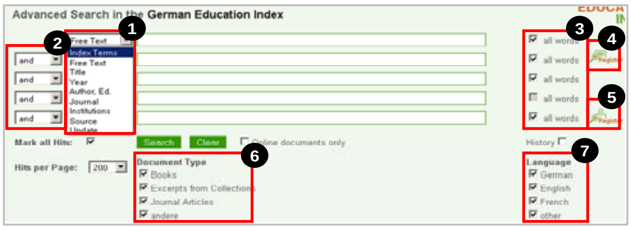


Figure 1: Quick search of the German Education Index

As illustrated in figure 2, the advanced search mode offers more sophisticated search functionalities than the quick search. The user can define searches in one or more of the different fields *free text, index terms, title, year, author/editor, journal, institutions, source* or *update* (1) and combine the search in different fields by the Boolean operators *AND, OR* and *NOT* (2). In every field, multiple terms are by default combined by *AND*, but a configuration by the user is possible (3). To define his/her query, the user can use an index term register (4) and a person register (5). Further restrictions are possible by document type (6) and by language (7).



and / and / and / and

☑ all words
☑ all words
☑ all words
□ all words
☑ all words

Mark all Hits: ☑

□ Online documents only

History □

Hits per Page: 200

Document Type
☑ Books
☑ Excerpts from Collections
☑ Journal Articles
☑ andere

Language
☑ German
☑ English
☑ French
☑ other

Figure 2: Advanced Search of the German Education Index

Capitalization is disregarded by the system. If a user enters a term that can be looked up in a synonym list, the synonym(s) is (are) automatically added to the query, connected by the OR-operator. Furthermore, the system provides a "Did you mean" – functionality. It is based on the Levenshtein algorithm and generates search suggestions if a query has invoked an empty result set. Depending on the search field, the entered query is compared to a register over this field. When a user searches for a last name that several authors have in common, the respective full author names are listed for specification, for example.

4 Method

To analyze users' search tactics and strategies, logfile analysis is a promising method [Jansen, 2009]. One of its advantages is the availability of large amounts of data that do not need to be collected especially for analysis purposes. Moreover, logfiles document real user queries and no artificial situation is created, such as in formal retrieval experiments.

Nevertheless, logfile analyses do not allow to get into direct contact with the users. Consequently, the users' information needs are unknown and can only be inferred from the available data.

For this study, the logfiles of one day, April 1st 2008, were analyzed. They were ordered both chronologically and by IP address. This way, the logfiles of each user were accumulated in separate text files.

In the next step, the files were filtered by size with the aim of restricting the analyses to files that were likely to comprise complex searches. For this purpose, a minimum size of 10 kb was defined, which was fulfilled by 153 files. The discarded 512 smaller files were not expected to comprise complex search processes that were in the focus of this study.

Afterwards, the text files were split up into search sessions whose distinct query steps were timely and thematically related. If the time between two user actions was more than 30 minutes and/or if a query clearly addressed a new topic, this was considered as an indicator for the start of a new search session. This way, 235 search sessions were identified. These may comprise several distinct queries, each of which may consist of one or more query terms.

The query reformulation steps throughout the search sessions could be described by term tactics. Partly based on the classification by [Bates, 1979] presented in section 2.2, several tactics were identified, as listed in figures 3 and 4. Whenever a term in a query was deleted, replaced, altered or added, these actions were classified as one of these tactics.

The term tactics ranged from semantic relations such as *broader terms, narrower terms, synonyms, related terms, antonyms* and *compounds* over the use of *affixes*, *singular* or *plural* forms and *truncations* to changes in *word order*, the use of *spacing* characters between terms, *phrase searches*, *translations* (e.g. from English to German) and *term conversions* (e.g. from adjective to substantive). If none of these tactics was applied, the action was classified as an unrelated term.

Term Tactics		
Broader Term	Compound	Translation
Narrower Term	Affix	Spacing
Synonym	Singular/plural	Phrase Search
Related Term	Truncation	Conversion
Antonym	Word Order	Unrelated Term

Figure 3: Identifiable term tactics

Search formulation tactics that could be identified by logfile analyses in this study refer to the use of Boolean operators, as illustrated in figure 4. If elements were added by one of the operators or if one such element was deleted, this was considered as a distinct search formulation tactic.

Search Formulation Tactics
Addition of an AND-element
Deletion of an AND-element
Addition of an OR-element
Deletion of an OR-element

Figure 4: Identifiable search formulation tactics

Out of the strategies presented in section 2.2, the *quick approach*, *briefsearch*, *building blocks* and *pairwise facets* could be identified by logfile analyses, as shown in figure 5.

Subject Search Strategies
Quick Approach
Briefsearch
Building Blocks Approach
Most Specific Query First
Most General Query First
Pairwise Facets

Figure 5: Identifiable subject search strategies

Unfortunately, the logfiles did not give information about the number of result documents for each query. For this reason, the *most specific concept first* and the *fewest postings first* strategies could not be identified. Instead, two new strategies were defined, inspired by Harter's strategy definitions. The *most general query first strategy* starts with a general query which is afterwards specified, for example by adding new facets with the Boolean AND-operator. The *most specific query first* strategy proceeds in the contrary way, starting with a specific query, consisting of several facets, and deleting or generalizing parts of the query step by step.

As the study focused on the analysis of a single search system, the *multiple briefsearch* strategy was not identifiable. Furthermore, the logfile analyses did not allow to deduce the consequences that users drew from the viewing of documents, which impeded the detection of *citation indexing strategies* and *interactive scanning*. As the result documents were not inspected in this study, the *citation pearl growing* approach could not be identified either.

5 Results

5.1 Search statistics

On average, the users reformulated their query 7,8 times per search session, which accounted for an average session length of 8,8 steps. The mean query length was 2,2 terms per query.

The major part of the search sessions comprised queries with an average query length between one and two terms, as shown in figure 6. It categorizes search sessions by their average query lengths. For example, a search session with several one-term queries would have an average query length of 1, while a session with two two-term queries and two three-term queries would have an average query length of 2,25. In 8,1% of the search sessions only one query term was used on average, in 50,6% of the search sessions, the average query length was between 1 and 2 terms, in 29,4% of the sessions it was between 2 and 3 terms, and in 11,9 % the average query length was higher than 3.

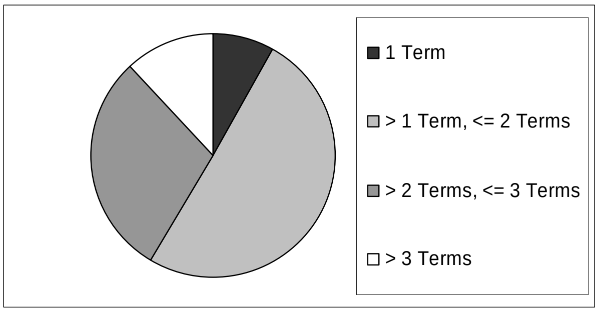


Figure 6: Percentage of sessions with an average query length of x terms

The majority of the 235 search sessions was conducted in the advanced search mode (78,3%), which is illustrated in figure 7. In 17,9% of the search sessions, a combination of the quick and advanced search masks was used and in another 3,8% of the sessions only the quick search was employed. This may be due to the fact that the analyses mainly focused on complex searches.

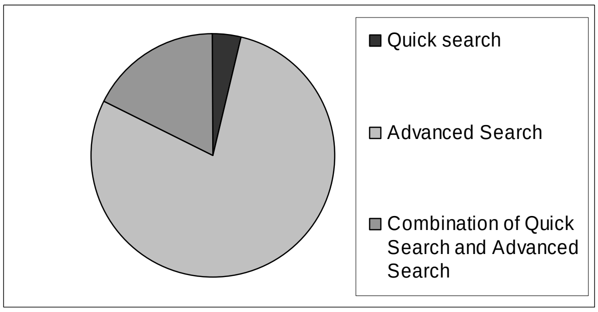


Figure 7: Search masks

If a users followed a link in the result set, he/she was presented the bibliographic data of a document, which is defined as a document view here. The average number of visited documents was only 2,4 per search session. In 11,91% of the search sessions, no result document at all was visited, in 43,4% of the sessions only one document was visited and in 34,89% of the searches, 2-5 documents were visited, which is illustrated in figure 8.

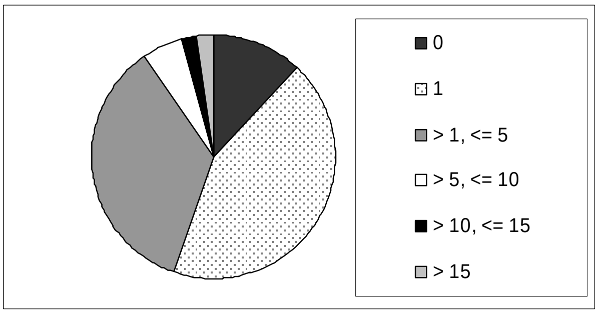


Figure 8: Percentage of sessions with x documents viewed

Rather than being directly consulted by the users, the registers were preferably used indirectly when the system provided register based search suggestions. These suggestions were accepted by the users in 113 cases whereas they autonomously consulted the available person and index term registers in only 14 cases. Unfortunately, the logfiles did not give evidence of the number of suggestions that were disregarded by the users. Users addressed the registers most frequently for looking up index terms (92,9%) while the consultation of the person register accounted for only 7,1% of the register uses. By contrast, the system generated suggestions accepted by the users were mainly based on the person register (62%).

5.2 Overview of search strategies

The following figure 9 illustrates how the strategies can be split up into subject search strategies and non-subject search strategies. On the whole, 440 strategies were identified in the 235 search sessions. Subject search strategies constituted the biggest part (56,8%), while non-subject search strategies made up for the remaining 43,2%.

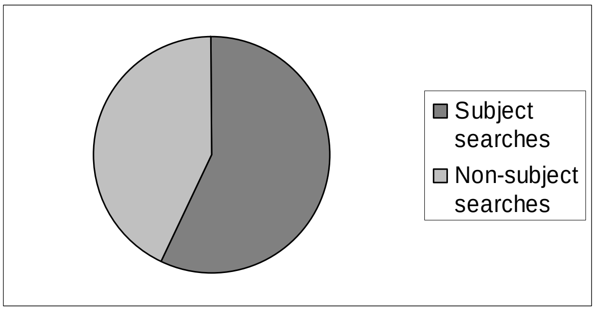


Figure 9: Percentage of subject and non-subject searches

Figure 10 shows how the 250 subject search strategies could be classified. The biggest amount of strategies were *briefsearches* (47,6%), followed by the *quick approach* (23,2%), the *pairwise facets* strategy (11,6%), the *most general query first* strategy (10,4%) and the *most specific query first* strategy (5,6%). In 1,2% of the cases, a combination of the two latter strategies could be identified. The *building blocks* approach was used in only one of the cases.

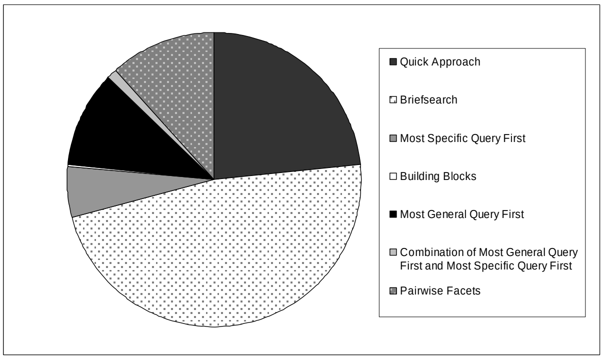


Figure 10: Subject search strategies

The following sections present examples for each of the identified strategies. Furthermore, they analyze which search tactics were applied in each of these search strategies.

5.3 Briefsearch

The following figure 11 shows how often each of the term tactics listed in section 2.2 was applied in the total amount of 119 *briefsearches*. Obviously, one of the most frequently used tactics was the replacement of a term with a completely *unrelated term* (24x). The second group of tactics was made up by thesaurus relations that were applied for search reformulation by the users: *related term*s were used in 18 of the cases, *broader term*s 11x, *narrower terms* 10x and *synonyms* 13x. Syntactic variants such as *plural or singul*ar forms (9x) and *truncations* (9x), *compounds* (7x), *translations* (6x), *conversions* (5x) and *phrases* (4x) made up the third group of tactics in *briefsearches*.

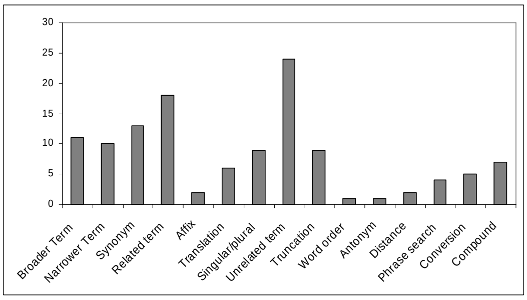


Figure 11: Briefsearch term tactics

The following figure 12 traces an example of a briefsearch. The example queries given for the distinct search strategies were originally defined in German language. To increase the readability of this paper, they were translated into English.

Step	Query	Field
1	„deaf” AND “language support”	index terms
2	“DEAF PERSON” AND “language support”	index terms
3	“deaf” AND “language development”	index terms

Figure 12: Example of a briefsearch

In this example, the user started with a Boolean AND search, consecutively replaced both initial query terms and tried out different term combinations and search parameters (AND, OR).

5.4 Quick Approach

In the 58 *quick approach* searches, fewer different term tactics could be identified than in the *briefsearches*. This is due to the fact that the *quick approach* searches comprised only one word queries in this study because query terms were by default combined by the AND operator.

Related terms made up the most frequently used term tactic in the *quick approach* (16x), followed by the use of *broader term*s (7x), *narrower term*s (4x), *synonyms* (4x) and syntactic variants such as *translations* (3x), *affixes* (1x) and *singular and plural* forms (1x). The remaining term tactics could not be detected in the *quick approach* searches.

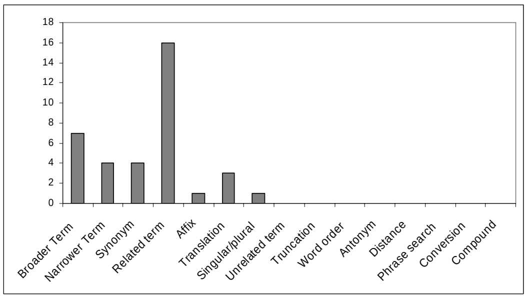


Figure 13: Quick approach term tactics

The following figure 14 gives a brief example for a *quick approach* where the user defined a simple one-term query:

Step	Query	Field
1	“school garden“	free text

Figure 14: Example of a quick approach

5.5 Pairwise Facets

In the 29 searches adhering to the *pairwise facet*s strategy, a big variety of different term tactics could be identified, such as in the *briefsearches*. But in contrast to the term tactics in *briefsearch*, the use of *related terms* accounted for the highest number of tactics in the *pairwise facets* strategy.

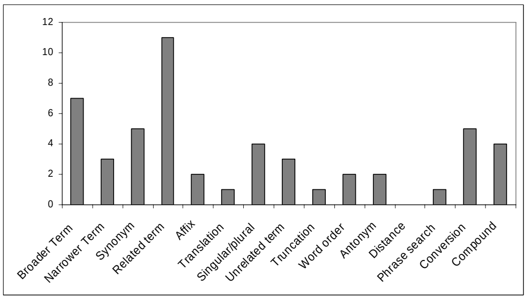


Figure 15: Pairwise facets term tactics

The following example in figure 16 illustrates the application of the *pairwise facets* strategy.

Step	Query	Field
1	“intercultural“ AND “competence“ AND “pedagogics“	index terms
2)	“competence“ AND “pedagogics“	index terms
3)	“intercultural“ AND “learning“	index terms
4)	“intercultural“ AND “competence“	index terms
5)	“intercultural“ AND “pedagogics“	index terms

Figure 16: Example of a pairwise facets strategy

In this example, the user started with a query consisting of the three facets “intercultural”, “competence” and “pedagogics”, which were consecutively combined in pairs in steps 2 to 5. This example also illustrates the tactic of using *related terms* (“pedagogics” and “learning”), which was identified as the most frequently applied tactic for this strategy.

5.6 Most General Query First

Term tactics in the 26 *most general query first* searches were dominated by the use of *narrower terms* (5x) and *broader term*s (4x). *Unrelated terms*, *spacing* characters, *compounds* and *singular* and *plural* forms were all applied in two of the cases.

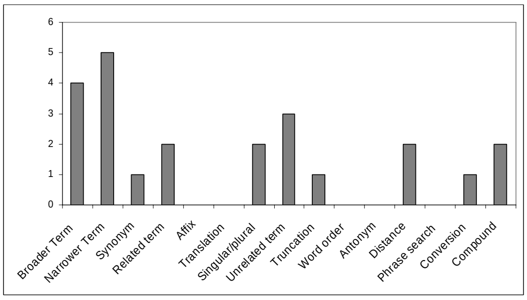


Figure 17: Most general query first term tactics

The following example in figure 18 illustrates the use of the most general query first strategy.

Step	Query	Field
1	“education“	index terms
2)	“education“ AND “fundamentals“	index terms
3)	“education“ AND “fundamentals“ AND “tasks“	index terms

Figure 18: Example of a most general query first strategy

In this example, the user started with a general term that was consecutively specialized with new facets that were combined with the AND operator. Starting with the general query “education”, the new facet “fundamentals” was added in step 2, followed by another facet in step 3.

5.7 Most Specific Query First

In the 14 *most specific concept first* searches, only few term tactics could be identified - the use of a *broader term* in one case, the use of *affixes* in one case and the use of a *conversion* in one case.

The following example in figure 19 illustrates the use of the most specific query first strategy.

Step	Query	Field
1)	“comparative tests”	free text
	AND “German“	index terms
	AND “class“	index terms
2)	“comparative tests”	free text
	AND “German“	index terms
3)	“comparative tests”	free text

Figure 19: Example of a most specific query first strategy

In this example, the user started with a very specific query and then deleted the different facets consecutively.

5.8 Combination of Most General Query First and Most Specific Query First

In three cases, users applied a combination of the *most general concept first* and the *most specific concept firs*t strategies. They either started with a very specific query which was then generalized and afterwards specialized again with different terms, or the other way round. An example of such a strategy combination is given in figure 20.

Step	Query	Field
1	“deprivation“	index terms
2)	“deprivation“ AND “children“	free text
3)	“deprivation“ AND “children“ AND “alcohol“	free text
4)	“infantile“ AND “deprivation“	free text
…		

Figure 20: Example of a combination of the most general query first and most specific query first strategies

In this excerpt of a search, the user started with a general query which was then consecutively specialized in steps 2 and 3 before generalizing it again in step 4.

5.9 Building Blocks

Although the *building blocks* approach could be identified only once, an example is given in figure 21.

Step	Query	Field
1	“method competence“	free text
2)	“method competence“	free text
	AND (“library“ OR “school library“	index terms

Figure 21: Example of a building blocks strategy

In this example, the user defined a query with two facets which were combined by the AND-operator. The second facet was expressed by a combination of two terms combined by the OR-operator.

5.10 Overview of Term Tactics

The following figure 22 gives an overview of the percentages of the different term tactics throughout all subject search strategies. It illustrates that *related terms* were most frequently used (19,1%), followed by the use of *broader terms* (17,5%), completely *unrelated terms* (12,2%), *narrower terms* (8,9%) and *synonyms* (9,4%). *Singular and plural* forms made up 6,5% of the term tactics, followed by *compounds* (5,3%), *truncations*, *conversions* (each 4,5%) and *translations* (4,1%). The tactics *affix*, *word order*, *antonym*, *spacing* and *phrase search* each had a share of less than 4%.

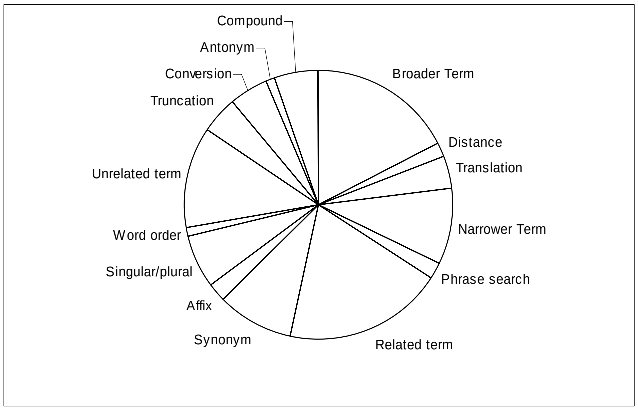


Figure 22: Percentage of term tactics used

5.11 Overview of Search Formulation Tactics

The use of search formulation tactics is illustrated in figure 23. It shows that combinations of search terms were most frequently defined by the Boolean AND operator, which was either added (47,6%) or deleted (48,4%). This may be due to the fact that the combination of search terms by the AND-operator was the default configuration. This configuration was changed very seldom to add OR-elements, which accounted for only 4% of the operator usages.

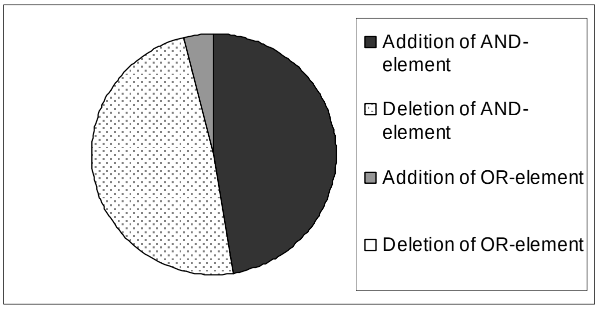


Figure 23: Percentage of search formulation tactics used

6 Conclusion and Outlook

This study focused on the identification of tactics and strategies in complex subject searches in the German Education Index. With the help of logfile analyses, the *quick approach, briefsearch, building blocks, most specific query first, most general query first* and *pairwise facet* approaches could be identified. *Briefsearches* and the *quick approach* were proven to make up for the biggest part of queries, even in the sample of mainly complex searches to which this study was restricted.

Nevertheless, more sophisticated strategies such as *building blocks, pairwise facets, most specific query first, most general query first* and a combination of the latter two could also be detected. Future research will have to investigate if these sophisticated strategies are also successful and effective in terms of recall and precision. If this is the case, the system may be designed to offer more support in the application of such strategies.

For deeper analysis in the future, the logged searches may be reconstructed. This way, the result lists can be analyzed in depth and the number of result documents can be determined so that strategies such as *lowest postings first* and *most specific concept first* may be identifiable.

Throughout the analyzed complex searches, the use of the advanced search mask was high. This means that users were aware of and familiar with the advanced search functionality. This is a main difference from the use of web search engines, as discussed in section 2.1. Nevertheless, the use of Boolean operators in the advanced search mode was rare as users tended to maintain the default configurations.

But they used a wide variety of term tactics throughout their searches, of which the semantic relations such as *broader terms*, *narrower terms*, *synonyms* and *related terms* made up the biggest part. While further investigations still need to verify if these tactics also prove to be effective, the use of such tactics can be supported by the system, such as by the suggestion of semantically related query expansion terms. In this context, the possibility of using an ontology for query expansion support in the German Education Index is currently examined [Carstens, 2009].

On the whole, users made frequent use of the suggestions the system already provided, which speaks for their general acceptance of search suggestions. The autonomous use of registers, by contrast, was very rare. Consequently, suggestions or possible expansion terms should be proactively offered to the user.

Compared to the users of web search engines, the users of the German Education Index made frequent reformulations in the analyzed searches. But it has to be stated that these focused on mainly complex searches. This also accounts for the comparatively high average session length of 8,8 query steps. The average query length of 2,2, by contrast, was even slightly lower than in the studies mentioned in section 2.1.

Throughout the search sessions, most users viewed less than 5 documents. These results indicate a tendency for selective searches in the analyzed sessions, while recall-oriented overview searches scarcely occurred. But it still needs to be examined if this also applies to the remaining less complex searches that were not inspected in this study. If this is the case, future enhancements of the retrieval system should focus on the support of precision-oriented processes.

While this study primarily examined subject searches, non-subject searches are planned to be analyzed in a similar way in the near future, shedding light on the way users search for documents based on already known facts.

References

[Bates, 1979] Marcia Bates. Information Search Tactics. *Journal of the American Society for Information Science*, 30(4):205–214, July 1979.

[Carstens, 2009] Carola Carstens. Effects of Using a Research Context Ontology for Query Expansion. In *Proceedings of the 6th European Semantic Web Conference*, pages 919-923. Springer.

[Chu, 2003] Heting Chu. Information Representation and Retrieval in the digital Age. Information Today, Medford, 2003.

[Harter, 1986] Stephen P. Harter. Search Strategies and Heuristics. In *Online Information Retrieval. Concepts, Principles and Techniques.* Academic Press, Orlando, Florida, 1986: 170–204.

[Jansen *et al.*, 2000] Bernard J. Jansen, Amanda Spink, and Tefko Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing and Management*, (36,2):207-227, January 2000.

[Jansen, 2009] Bernard J. Jansen. The Methodology of Search Log Analysis. In: Bernard B. Jansen, Amanda Spink, and Isak Taksa: *Handbook of Research on Web Log Analysis.* Information Science Reference, Hershey, USA, 2009: 100–123.

[Rieh and Xie, 2006] Soo Young Rieh, and Hong Xie. Analysis of multiple query reformulations on the web: The interactive information retrieval context. In *Information Processing and Management,* 42(3):751-768, 2006.

[Silverstein and Henzinger, 1999] Craig Silverstein, Hannes Marais, Monika Henzinger, Michael Moricz. Analysis of a Very Large Web Search Engine Query Log. *SIGIR Forum*, 33(1):6-12, 1999.

[Spink *et al.*, 2001] Amanda Spink, Dietmar Wolfram, B. J. Jansen, and Tefko Saracevic. Searching the Web: The Public and Their Queries. *Journal of the American Society for Information Science and Technology*, 52(3):226–234, February 2001.

[Sutcliffe *et al*., 2000] A. G. Sutcliffe, M. Ennis, and S. J. Watkinson. Empirical Studies of End-User Information Searching. *Journal of the American Society for Information Science*, 51(13):1211-1231, November 2000.

[Wildemuth and Moore, 1995] Barbara M. Wildemuth, and Margarte E. Moore. End-user search behaviors and their relationship to search effectiveness. *Bulletin of the Medical Library Association*, 83(3):294-304, July 1995.

Das LFRP-Framework zur Beherrschung komplexer Suchsituationen

Raiko Eckstein, Andreas Henrich, Nadine Weber
Otto-Friedrich-Universität Bamberg
Lehrstuhl für Medieninformatik
96052, Bamberg, Deutschland
{raiko.eckstein, andreas.henrich, nadine.weber}@uni-bamberg.de

Abstract

Der Bedarf an unterschiedlichen „Informationsträgern“, nachfolgend als Artefakte bezeichnet, sowie eine unklare Vorstellung über die benötigte Information beim Nutzer resultieren in komplexen Suchsituationen, wie sie bspw. im Bereich der Entwicklung technischer Produkte zu finden sind. Hier werden unterschiedlichste Artefakte wie Dokumente, Produkte oder Materialien im Produktentwicklungsprozess benötigt, wobei Informationsbedürfnisse oft nur vage und ungenau beschrieben werden können. Zusätzlich erschweren die Heterogenität der vorhandenen Archivierungssysteme und das Fehlen einer übergreifenden Suchfunktion das Auffinden relevanter Information. Aus diesem Grund schlagen wir mit unserem LFRP-Framework ein interaktives Retrievalmodell zur Beherrschung derartig komplexer Suchsituationen vor. LFRP steht für die vier Basiselemente, die zu einem übergreifenden Framework integriert werden. Durch die Verwendung mehrerer Artefaktebenen (Layer) wird der Vielfalt nachgefragter Artefakte begegnet. Die Suche nach den Artefakten selbst erfolgt gemäß dem Paradigma der Facettierten Suche. Allerdings erlaubt eine reine Filterung der Ergebnismenge keine Aussage hinsichtlich der Relevanz der Ergebnisse. Folglich erweitern wir das bekannte Konzept um Möglichkeiten zur Erstellung eines Rankings auf Basis sowohl von Facettenwerten als auch Query-by-Example (QbE)-Ansätzen. Zusätzlich schlagen wir eine visuelle Form der Anfrageformulierung mittels Paralleler Koordinaten vor, die Einblicke in die Charakteristika und Abhängigkeiten der gefundenen Ergebnisse bietet.

1 Motivation und Problemstellung

Um die Wettbewerbsfähigkeit von Unternehmen zu sichern, sind innovative Produkte in kurzer Zeit auf den Markt zu bringen. Dies betrifft insbesondere auch den Entwicklungsprozess im Maschinenbau, der im Forschungsverbund FORFLOW von Projektpartnern aus Industrie und Forschung betrachtet wird und den Hintergrund dieser Arbeit bildet. Zusätzlich werden Produkte durch die Einbindung elektrischer, elektronischer und computergesteuerter Komponenten als auch durch zunehmend dynamische Kundenanforderungen immer komplexer [Schichtel, 2002]. Dies spiegelt sich auch im Produktentwicklungsprozess (PEP) wider, in dem viele verschiedene Artefakte erzeugt und benötigt werden. Neben Produktdaten und Dokumenten sind auch projektspezifische Informationen, Materialdaten und Informationen über Lieferanten oder Experten von Nutzen für den Entwickler. Allerdings nimmt die Suche nach relevanten Informationen laut [Grabowski und Geiger, 1997] ca. 20 - 30% der gesamten Entwicklungszeit in Anspruch. Dies ist v. a. darauf zurück zu führen, dass die während des PEP erstellten und benötigten Daten in unterschiedlichen Systemen archiviert werden. Neben Product Data Management (PDM)-Systemen, kommen Enterprise Resource Planning (ERP)-Systeme, Document Management-Systeme (DMS), Datenbanken (DB) und selbst einfache Dateisysteme zum Einsatz. Obwohl diese Systeme Suchfunktionen anbieten, finden Entwickler häufig nicht die benötigte Information. Dies liegt zum einen daran, dass Entwickler oft selbst nicht genau beschreiben können, wonach sie eigentlich suchen. Zum anderen ist häufig aufgrund einer fehlenden übergreifenden Suchfunktionalität nicht bekannt, dass eine Information existiert oder in welcher Quelle sie enthalten ist.

In der Literatur wurden bereits verschiedene Möglichkeiten zur Verbesserung der Informationsversorgung von Produktentwicklern erforscht (vgl. Kapitel 2). Jedoch handelt es sich hierbei größtenteils um Ansätze, die nur ein spezielles Informationsbedürfnis adressieren. Diese Einzellösungen sind zwar in manchen Situationen durchaus hilfreich; jedoch sind sie nicht für die Befriedigung komplexer Informationsbedürfnisse geeignet. Befindet sich ein Entwickler z. B. in einer frühen Phase des PEP, in der er ein Lösungskonzept für definierte Produktanforderungen erstellen muss, so interessieren ihn in dieser Situation eher Lösungskonzepte aus früheren Projekten, die mit ähnlichen Anforderungen verbunden waren. In späteren Phasen wie z. B. der eigentlichen Produktkonstruktion hingegen, suchen Entwickler vorrangig nach verwendbaren CAD-Modellen oder nach Informationen über Materialeigenschaften.

Folglich wird ein übergreifendes interaktives Retrievalmodell benötigt, das unterschiedliche Artefakte aus verschiedenen Quellsystemen einbezieht und den Nutzer in zweierlei Weise unterstützt. Zum einen wird eine zielorientierte Suche gemäß dem Konzept der *Known-Item Search* [Reitz, 2004] benötigt, die in Situationen, in denen der Produktentwickler eine definierte Vorstellung über sein Suchobjekt besitzt, dieses zielgerichtet bereitstellt. In Situationen hingegen, in denen der Entwickler nicht genau weiß wonach er sucht, ist eine explorative Vorgehensweise erforderlich, bei der der Anfragende durch *Browsen* des Datenbestandes auf Informationen stößt, die für sein Informationsbedürfnis relevant sind [Marchionini, 2006].

Erste Ideen hierzu wurden zuerst in [Eckstein und Henrich, 2008] skizziert und in [Eckstein und Henrich, 2009] mit dem Fokus auf der Anfragekomponente und der Umsetzung der graphischen Benutzeroberfläche vertieft. Das vorliegende Papier präsentiert einen Gesamtüberblick über das entwickelte Framework, indem es sowohl die Indexierungs- als auch die Anfrageseite berücksichtigt. Nach einem Überblick über bereits existierende Retrievalansätze stellen wir in Kapitel 3 unser *LFRP-Framework* für komplexe Suchsituationen anhand einer Beispielanfrage im Prototyp unserer graphischen Benutzeroberfläche (GUI) vor. Ausgehend von der Grundidee einer facettierten Suche erläutert Kapitel 4 im Anschluss die für ein derartiges Suchframework notwendige Indexierung, die Artefaktbeschreibungen auf Basis von Artefakttyphierarchien aus diversen Informationsquellen generiert (Kap. 4.1) und dabei Beziehungen zwischen Artefakten berücksichtigt (Kap. 4.2). Im Anschluss beschäftigt sich Kapitel 5 detaillierter mit der Suchseite des Frameworks. Dabei werden die visuelle Anfrageformulierung mittels paralleler Koordinaten (||-coords), die Integration von Ranking- und QbE-Kriterien (Kap. 5.2 und 5.3), sowie die Nutzung von Artefaktbeziehungen zur Verbesserung der Suchfunktionalität mit Hilfe eines Ebenenkonzeptes (Kap. 5.5) erklärt.

2 State of the Art

In den letzten beiden Jahrzehnten beschäftigten sich bereits viele Forscher mit der Herausforderung, die Informationsversorgung von Produktentwicklern im Speziellen, aber auch innerhalb von Unternehmen im Allgemeinen zu verbessern. Dabei sind verschiedene Lösungsansätze entstanden, die im Folgenden skizziert werden.

Da das Auffinden ähnlicher, bereits existierender Produkte wesentlich dazu beiträgt, redundante Tätigkeiten zu vermeiden und so sowohl Entwicklungszeiten als auch -kosten zu reduzieren, bildete sich ein wichtiger Forschungsschwerpunkt im Bereich der Ähnlichkeitssuche. Aufgrund der zunehmenden Verfügbarkeit von 3D CAD-Modellen konzentrierte man sich hierbei hauptsächlich auf die Entwicklung von Retrievalansätzen, welche einen Vergleich von Produkten auf Basis ihrer dreidimensionalen Geometriebeschreibungen ermöglichen. Somit sind in der Literatur unterschiedliche Möglichkeiten zur Repräsentation dieser Geometriemodelle zu finden, die u. a. auf Tiefenpuffer-Informationen [Vranic, 2004], Abstandsverteilungen beliebiger Objektoberflächenpunkte [Osada *et al.*, 2002] oder Skelettgraphen [Sundar *et al.*, 2003] beruhen. Des Weiteren wurden derartige Anstrengungen auch für technische Zeichnungen unternommen, zu denen wir in [Weber und Henrich, 2007] einen Überblick gegeben haben. Obwohl oder vielleicht gerade weil diese Ansätze nur einzelne, spezielle Informationsbedürfnisse fokussieren, konnten sich einige Konzepte – v. a. im 3D-Bereich – auch kommerziell durchsetzen. Beispiele sind die geometrische Suchmaschine für 3D-Daten *Geolus Search*[1] der Siemens AG oder *CADFind*[2] der Applied Search Technology Ltd.

Des Weiteren hat man versucht, derartige Einzellösungen in umfassendere Ansätze zu integrieren. Ein Beispiel hierfür ist das *Design Navigator System* von Karnik et al. [2005], welches zur Unterstützung der Produktentwicklung im militärischen Bereich prototypisch entwickelt wurde. Neben Möglichkeiten zur Suche nach Produktinformationen (Anfragemöglichkeiten hinsichtlich der Geometriebeschreibung, Produktfunktion und Produktstruktur), wurde hierbei insbesondere die Designhistorie einer Produktentwicklung bei der Suche mit berücksichtigt, um so v. a. neue und unerfahrene Entwickler bei der Arbeit zu unterstützen.

Obwohl alle diese Ansätze interessante Möglichkeiten der Unterstützung bieten, fand bisher kein Konzept breite Anwendung in der Praxis. Stattdessen ist die unternehmensinterne Softwarelandschaft durch heterogene Systeme geprägt, die jeweils nur bestimmte Kernaspekte fokussieren. Während bei PDM/PLM[3]- und ERP-Systemen das Produkt mit all seinen Informationen im Vordergrund steht, wird der Fokus bei DMS auf Dokumente und bei CRM[4]- und SRM[5]-Systemen auf Kunden- und Lieferantendaten gelegt. Dabei verfügt jedes einzelne System standardmäßig über Suchfunktionalitäten, die sich allerdings eher am Konzept der in Kapitel 1 genannten Known-Item Search orientieren.

Um dieser Heterogenität von Insellösungen und damit der Begrenztheit von verfügbaren Informationen zu begegnen, wurden in den letzen Jahren immer mehr Enterprise Search-Lösungen entwickelt. Durch die Nutzbarmachung des gesamten in einem Unternehmen verfügbaren Wissens, ist es Ziel dieser Systeme eine unternehmensweite Informationsbereitstellung zu gewährleisten.

Obwohl derartige Systeme neben allgemein gültigen Artefakten wie E-Mails oder Office-Dokumenten auch spezielle Artefakttypen wie CAD-Modelle, technische Zeichnungen, usw. berücksichtigen, werden diese vorrangig textuell analysiert. Dieser Ansatz reicht jedoch gerade bei speziellen Dokumenttypen nicht aus, da wichtige Informationen eben nicht nur in der Textform, sondern v. a. in graphischen Beschreibungen enthalten sind.

3 LFRP-Framework für komplexe Suchsituationen

Für die Realisierung sowohl eines zielgerichteten als auch eines explorativen Suchvorgehens stützen wir unser LFRP-Framework auf das Konzept der *facettierten Suche*, welches den Nutzer durch Selektionskriterien, sog. Facetten, bei seiner Anfrageformulierung unterstützt [Yee *et al.*, 2003]. Eine Facette beschreibt einen Aspekt eines Artefakts (z. B. Erstellungsdatum bei Dokumenten) und kann während der Indexierung ermittelt werden. Die facettierte Suche stellt dem Nutzer Anfragevorschauen zur Verfügung. Diese Vorschauen bestehen aus der Anzahl der Artefakte, die im Ergebnis verbleiben würden, wenn der Nutzer die aktuelle Anfrage mit der betrachteten Facettenausprägung weiter einschränkt. So wird wirksam verhindert, dass der Nutzer durch zu eng gestellte Filterungen eine leere Ergebnismenge zurückgeliefert bekommt, da Facettenausprägungen, für die es bei der aktuellen Selektion keine Artefakte gibt, ausgeblendet werden. Die Registerkarte *Facets* im oberen Bereich unseres Prototyps (vgl. Abbildung 1) gibt eine Übersicht über die Facetten, indem unter *Add facet* eine dynamisch aktualisierte Liste von wählbaren Facetten angeboten wird. Aus dieser Liste kann der Nutzer

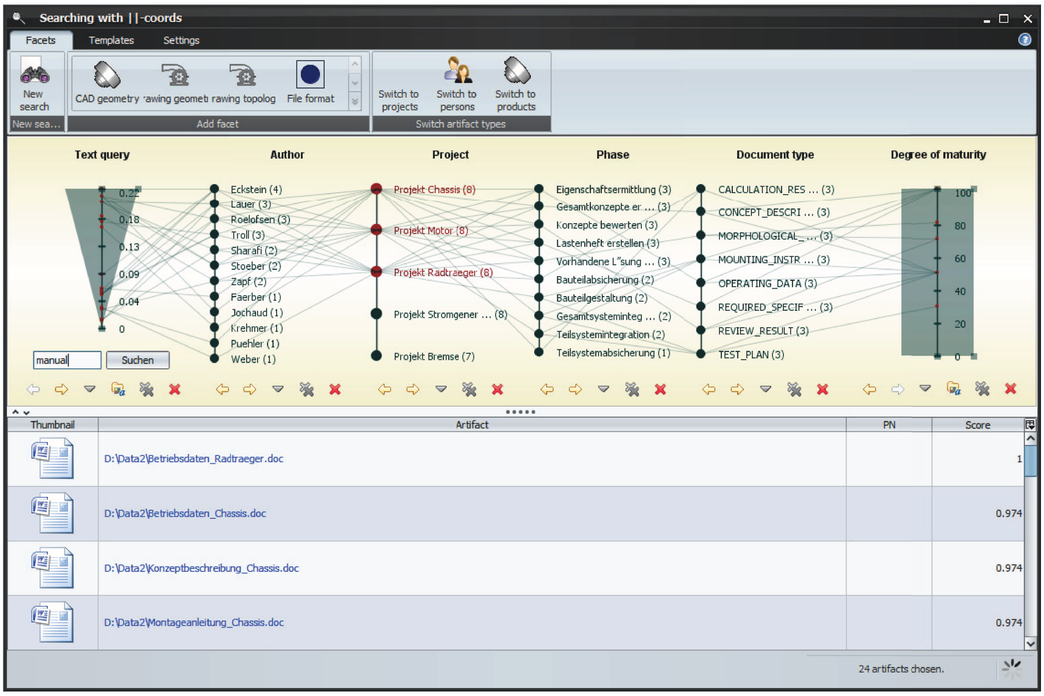
[1] http://www.plm.automation.siemens.com/de_de/products/open/geolus/index.shtml

[2] http://www.sketchandsearch.com

[3] Product Lifecycle Management

[4] Customer Relationship Management

[5] Supplier Relationship Managment



manual

Thumbnail	Artifact	PN	Score
	D:\Data2\Betriebsdaten_Radtraeger.doc		1
	D:\Data2\Betriebsdaten_Chassis.doc		0.974
	D:\Data2\Konzeptbeschreibung_Chassis.doc		0.974
	D:\Data2\Montageanleitung_Chassis.doc		0.974

24 artifacts chosen.

Abbildung 1: Screenshot der graphischen Benutzeroberfläche mit Darstellung einer Beispielanfrage.

die für seine Anfrage relevanten Facetten auswählen und so seine Anfrage schrittweise verfeinern.

Die Darstellung der gewählten Facetten erfolgt dabei im mittleren Bereich der GUI. Im Gegensatz zu herkömmlichen facettierten Suchansätzen verwenden wir hierfür einen visuellen Ansatz mittels ||-coords. Gemäß Inselberg [1985] wird jede Facette als eine vertikale Achse dargestellt, die die Facettenausprägungen als Achsenpunkte beinhaltet. Folglich wird eine beliebige Anzahl von Facetten durch parallele Achsen im Zusammenhang abgebildet, was die Veranschaulichung einer komplexen, aus mehreren Unteranfragen bestehenden Anfrage in einer übersichtlichen Form ermöglicht. Abbildung 1 zeigt eine Beispielanfrage nach Dokumenten unter Verwendung der Facetten *Text query*, *Author*, *Project*, *Phase*, *Document Type* und *Degree of maturity*. Dabei unterscheiden wir zum einen Attributfacetten, die die Auswahl einer oder mehrerer Facettenausprägungen zulassen und somit als reine Filterkriterien dienen (vgl. Facette *Project*: hier interessieren nur Dokumente, die in mindestens einem der drei im oberen Bereich der Achse gelisteten Projekte erstellt oder benötigt wurden). Diese Filterung bewirkt allerdings nur eine Einschränkung der Ergebnismenge, so dass für den Nutzer keine Rangfolge der Ergebnisse hinsichtlich ihrer Relevanz zur Anfrage ersichtlich ist. Aus diesem Grund wurden Präferenzfunktionen in das Retrievalmodell integriert, mit deren Hilfe der Nutzer ein Ranking der Ergebnisse anhand seiner Präferenzen für bestimmte Facettenwerte oder -wertebereiche ermitteln kann. Abbildung 1 zeigt eine solche Präferenzfunktion für die Facette *Degree of maturity*, bei der jeder Facettenwert im Intervall von 0 bis 100 mit dem gleichen Präferenzwert belegt wird. Obwohl der Nutzer hier prinzipiell jede beliebige Präferenzfunktion definieren kann, stellt das System für gängige Funktionen Vorlagen zur Verfügung (vgl. Kapitel 5.2).

Zusätzlich ermöglicht unser Framework die Verwendung von Ähnlichkeitskriterien für die Anfrageformulierung. Demzufolge können auch Ansätze zum Vergleich nicht nur der textuellen Ähnlichkeit sondern auch z. B. der 3D-Geometrieähnlichkeit in einem derartig übergreifenden Framework berücksichtigt und angewendet werden. Wie Abbildung 1 bei der Facette *Text query* zeigt, werden diese Ähnlichkeitsfacetten ebenfalls als Achsen dargestellt. Allerdings veranschaulichen ihre Achsenpunkte die ermittelten Retrievalstatuswerte, weshalb sie defaultmäßig mit einer Präferenzfunktion überlagert sind. Eine detaillierte Erläuterung hierzu wird in Kapitel 5.3 gegeben.

Die Ergebnismenge, die durch jede Aktion des Nutzers unmittelbar verändert wird, wird auf zwei Arten visualisiert. Zum einen wird jedes Artefakt in Form eines Linienzuges zwischen den parallelen Koordinaten dargestellt. Dies gibt sowohl Einblicke in die Charakteristika der Ergebnisse als auch in die Abhängigkeiten zwischen den gewählten Facetten. Zum anderen wird die Ergebnismenge im unteren Bereich der GUI zusammen mit einer etwas detaillierteren Beschreibung aufgelistet. Diese Beschreibung besteht aus einem Identifizierungsschlüssel (z. B. dem Dokumentpfad), evtl. einem Retrievalstatuswert (sofern verfügbar), der die Relevanz eines Ergebnisses angibt, und einem Vorschaubild, falls vorhanden.

4 Generierung spezifischer Artefaktbeschreibungen

Zur Realisierung des in Kapitel 3 vorgestellten LFRP-Frameworks wird eine Indexierungskomponente benötigt, die eine Beschreibung der Suchobjekte auf Basis von Facetten generiert. Dabei ist zu berücksichtigen, dass in komplexen Suchsituationen diverse Artefakte von Bedeutung sein können. So werden gerade in der Produktentwicklung nicht nur Dokumente, sondern v. a. auch Produktdaten und Materialinformationen nachgefragt. Für jeden dieser Artefakttypen sind andere charakteristische Facetten für die Suche relevant. Während bspw. für Dokumente der Dokumenttyp, das Dateiformat oder das Erstellungsdatum wichtige Filterkriterien darstellen, sind Produkte durch eine Sachnummer, ihre Produktgruppe, ihr Gewicht und an-

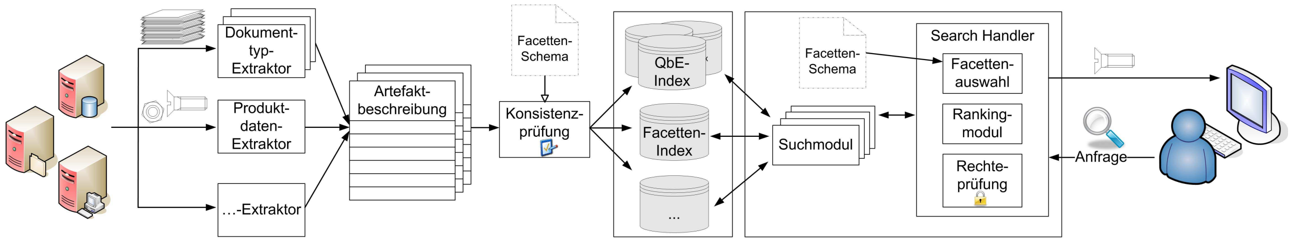


Abbildung 2: Architektur des LFRP-Frameworks.

dere beschreibende Eigenschaften charakterisiert. Folglich sind bei der Indexierung typ-spezifische Artefaktbeschreibungen zu erstellen. Abbildung 2 zeigt in der linken Hälfte den prinzipiellen Ablauf der Indexierung.

In einem ersten Schritt sind alle Informationen aus den im Unternehmen existierenden Verwaltungssystemen und Anwendungen zu sammeln. Das umfasst zum einen die Indexierung sämtlicher Dokumente aus DMS, DB und Dateiverzeichnissen. Aber auch Produktdaten aus PDM- oder ERP-Systemen, Informationen aus Materialdatenbanken und Projektmanagementsystemen, sowie Personendaten aus CRM-/SRM-Systemen und internen Organisationsstrukturen sind bei der Indexierung mit aufzunehmen. Die Indexierung selbst sollte dabei sowohl in regelmäßigen Zeitabständen, als auch beim Einstellen neuer oder modifizierter Artefakte erfolgen.

Für jedes der erfassten Artefakte ist in einem zweiten Schritt eine geeignete Artefaktbeschreibung zu generieren. Diese kann einerseits Repräsentationen wie z. B. Featurevektoren oder Histogramme enthalten, die einen Ähnlichkeitsvergleich des Artefakts mit anderen Artefakten des gleichen Typs auf Basis z. B. der Geometrie oder Topologie erlauben (im Weiteren als QbE-Repräsentationen bezeichnet). Zum anderen besteht sie aus Attributfacetten, die für das jeweilige Artefakt charakteristisch sind. Dabei ist zu beachten, dass ein Artefakt durchaus auch Informationen über andere Artefakte beinhalten kann. Insbesondere Dokumente enthalten oft zusätzliche Informationen wie z. B. den oder die Autoren des Dokuments. Somit können prinzipiell sowohl dokument- als auch personenspezifische Facetteninformationen extrahiert werden. Darüber hinaus dienen Dokumente v. a. in der Produktentwicklung dazu, Produkte näher zu beschreiben (z. B. durch CAD-Modelle, technische Zeichnungen, Stücklisten, . . .). Dies ermöglicht ein Auslesen von Produktfacetten oder – wenn vorhanden – auch von materialspezifischen Informationen. Da es neben diesen sog. Produktmodellen noch andere Dokumentarten wie Projektdokumente und allgemeine Dokumente (z. B. Richtlinien, Normen) gibt, sind abhängig vom Dokumenttyp Beschreibungen für jeweils alle in einem Dokument adressierten Artefakte zu erstellen.

Demzufolge werden für jeden Artefakttyp eine oder mehrere spezifische Extraktorkomponenten benötigt, die geeignete Artefaktbeschreibungen erstellen. Dabei stellt sich die Frage, wann eine Artefaktbeschreibung geeignet ist. Hierfür empfehlen wir im Rahmen der Indexierung die Durchführung einer Konsistenzprüfung, die in einem dritten Schritt die erstellten Artefaktbeschreibungen validiert und verifiziert. Dies erfolgt auf Basis eines Facettenschemas, das aus zwei Definitionsabschnitten besteht. Zum einen sind alle zu berücksichtigenden Facetten zusammen mit ihrem Facettentyp und ihrer Kardinalität (einwertig vs. mehrwertig) festzulegen (siehe Kapitel 5). Zum anderen enthält dieses Schema eine Artefakttyphierarchie, die beschreibt, welche Facetten für einen bestimmten Artefakttyp verfügbar sind. Auf Basis dieser beiden Informationen ist es schließlich möglich zu prüfen, ob eine Artefaktbeschreibung den Vorgaben des Schemas entspricht. Eine genauere Erläuterung hierzu geben wir im folgenden Kapitel 4.1.

Zusätzlich muss bei dieser Prüfung berücksichtigt werden, dass eine Information aus mehreren Quellen extrahiert werden kann. Beispielsweise sind die aus Produktmodellen extrahierten Produktinformationen (wie z. B. der Produktname) häufig in PDM-Systemen vorhanden. Obwohl die Produktinformation im Dokument normalerweise mit der im PDM-System übereinstimmen sollte, ist dies nicht immer der Fall, weshalb ein Abgleich der Informationen stattfinden muss. Gleiches gilt z. B. auch, wenn Informationen geändert werden und somit zu aktualisieren sind. Für die Beseitigung möglicher Unstimmigkeiten empfehlen wir die Definition von Regeln, die z. B. auf Basis der Extraktionsquelle oder des Extraktionszeitpunktes festlegen, welche Information aus welcher Quelle bei der Indexierung eines Artefakts zu verwenden ist.

Nach einer erfolgreichen Konsistenzprüfung können die Artefakte schließlich im Index gespeichert werden. Hierzu empfehlen wir die Unterteilung des Index in Subindizes, die die unterschiedlichen Aspekte der Artefaktbeschreibung enthalten. Demzufolge werden die Artefaktfacetten in einem Facettenindex und die QbE-Repräsentationen in entsprechenden QbE-Indizes abgelegt.

4.1 Integration von Artefakttyphierarchien

Abbildung 3 zeigt ein einfaches Beispiel einer Artefakttyphierarchie in UML-Notation, bei der Artefakte zunächst in die Typen *Material*, *Person*, *Produkt*, *Dokument* und *Projekt* unterteilt werden. Jeder einzelne Artefakttyp besitzt dabei spezifische Facetten. So werden z. B. Materialien durch ihren Materialschlüssel, einen Materialnamen, die zugehörige Materialgruppe, ihre Materialdichte und andere Facetten beschrieben. Eine Person hingegen besitzt einen Vor- und Nachnamen, eine Kontaktadresse und eine Rolle (z. B. Mitarbeiter, Lieferant oder Kunde). Derartige in den Blattknoten der Hierarchie befindliche Facetten gelten somit nur für Artefakte des jeweiligen Typs. Facetten des Wurzelknotens hingegen werden an alle Unterklassen vererbt. Demzufolge wird jedes Artefakt durch drei Parameter spezifiziert: eine ArtefaktID zur eindeutigen Identifizierung, ein Artefaktpfad, der angibt, wo das Artefakt verwaltet wird (z. B. Pfadangabe ins PDM für Produkte oder ins DMS für Dokumente) und ein (gemäß UML) „virtueller“ Artefakttyp, der die Spezialisierung der Artefakte auf oberster Ebene definiert.

Darüber hinaus zeigt Abbildung 3, dass auch Artefakte desselben Typs unterschiedliche Facetten besitzen können. Während bspw. Produkte wie Schrauben anhand ihrer Gewindeausführung, ihres Nenndurchmessers oder ihrer Schraubenlänge klassifiziert werden, interessieren bei

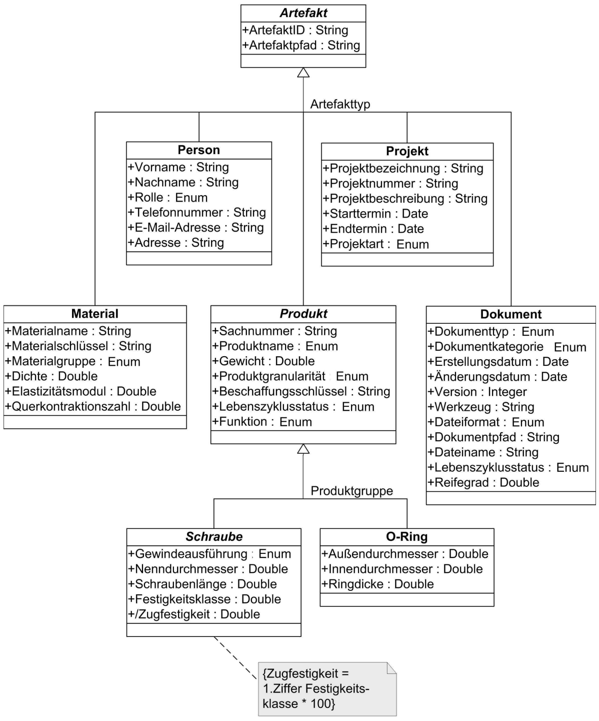


Abbildung 3: Beispiel einer einfachen Artefakthierarchie mit den „spezialisierenden“ Facetten Artefakttyp und Produktgruppe.

O-Ringen eher Werte für Außen- und Innendurchmesser. Somit hat jede einzelne Produktgruppe selbst wieder eigene charakteristische Facetten, was in einer eigenen Produkthierarchie resultiert.

Das bedeutet, dass wir grundsätzlich zwischen Facetten unterscheiden müssen, die eine Spezialisierung von Artefakten definieren, und Facetten die dies nicht tun. Dabei stellen die „spezialisierenden“ Facetten (Artefakttyp, Produktgruppe, ...) sozusagen Muss-Facetten dar, die in einer Artefaktbeschreibung enthalten sein müssen. Nur mit ihrer Hilfe kann die Indexierungskomponente bestimmen, welcher Subtyp in der Hierarchie betrachtet werden muss. Fehlen diese Informationen, sollten sie entweder aus anderen Quellen bezogen oder vom Nutzer erfragt werden, um einen eventuellen Informationsverlust zu vermeiden. Zusätzlich ist eine zweite Differenzierung von Facetten zu beachten. Zum einen gibt es Facetten bzw. Facettenwerte, die direkt aus der Quelle extrahiert werden können, wie z. B. das Erstellungsdatum eines Dokumentes. Zum anderen sind häufig aber auch abgeleitete Facetten, deren Wert von anderen Facetten und ihren Ausprägungen abhängt, als Selektionskriterien notwendig. So lässt sich bspw. die Zugfestigkeit einer Schraube aus ihrer Festigkeitsklasse ermitteln (vgl. Einschränkung in Abbildung 3). Indem Berechnungsvorschriften bei der jeweiligen Facettendefinition im Facettenschema mit angegeben werden, können derartige Abhängigkeiten berücksichtigt werden.

4.2 Beziehungen zwischen Artefakten

Wie in Kapitel 4 bereits angesprochen, kann ein Artefakt nicht nur Informationen über sich selbst, sondern auch über andere Artefakte beinhalten. Abbildung 4 zeigt beispielhaft verschiedene Artefakttypen und deren Beziehungen auf.

In Domänen wie der Produktentwicklung können derartige Beziehungsnetzwerke sehr komplex sein. Hier werden Projekte initiiert, um neue Produkte zu entwickeln oder um bestehende Produkte anzupassen. Bei einem Produkt handelt es sich entweder um ein Einzelteil oder um eine Baugruppe, die selbst wieder aus mehreren Produkten besteht. Die Durchführung der Projekte erfolgt durch Personen, die in den einzelnen Prozessphasen unterschiedlichste Dokumente erstellen und ändern. Unter anderem gehören hierzu Produktmodelle, die das zu entwickelnde Produkt näher beschreiben. Des Weiteren wird zur späteren Herstellung des Produktes ein bestimmtes Material festgelegt, dessen Eigenschaften den Anforderungen des Produktes genügen und das eventuell von diversen Lieferanten geliefert wird.

All diese Beziehungen tragen wertvolle Informationen, die bei der Befriedigung komplexer Informationsbedürfnisse zu berücksichtigen sind. So sollten bspw. Anfragen wie „*Finde alle zu einem Produkt verfügbaren Dokumente.*“ oder „*Finde alle Projekte, in denen für das entwickelte Produkt Material Z von Lieferant XY verwendet wurde.*“ bearbeitet werden können. Hierzu ist es erforderlich, die Artefaktbeschreibung um Beziehungen zu erweitern, die in einem separaten Index verwaltet werden. Die Definition möglicher Beziehungen zwischen Artefakten erfolgt dabei ebenfalls innerhalb der Artefakttyphierarchie, so dass sie u. a. bei der Erstellung mehrerer Artefaktbeschreibungen aus einer Quelle gesetzt werden können. Die Suchseite kann anschließend diese Beziehungen im Rahmen eines Ebenenkonzepts ausnutzen, um so die Suchfunktionalitäten für den Nutzer zu erweitern und damit die Retrievalqualität zu verbessern (vgl. Kapitel 5.5).

5 Anfrageformulierung mit ||-coords

Die rechte Hälfte von Abbildung 2 stellt den grundsätzlichen Aufbau des LFRP-Anfrageframeworks dar und illustriert den Ablauf einer Suchanfrage. Hierbei ist insbesondere hervorzuheben, dass die Anfragestellung interaktiv erfolgt, d. h. der Nutzer muss nicht die gesamte Anfrage formulieren, bevor er diese an das System stellt. Der Nutzer hat die Möglichkeit, Unteranfragen in Form von weiteren Facettenselektionen zur aktuellen Anfrage hinzufügen bzw. vorhandene Unteranfragen zu modifizieren. Stellt sich heraus, dass eine durch eine Unteranfrage vorgenommene Filterung nicht zielführend ist, kann diese Filterung modifiziert oder entfernt werden. Die Darstellung wird nach jeder Änderung der Suchanfrage angepasst, d. h. die aktuell gültigen Facettenausprägungen und die Linienzüge werden neu berechnet, sowie die Suchergebnisliste angepasst, um nur die aktuell gewählten Artefakte zu präsentieren.

Bei jeder Änderung der Anfrage wird diese an das Suchframework übergeben. Das zentrale Modul ist der *Search*

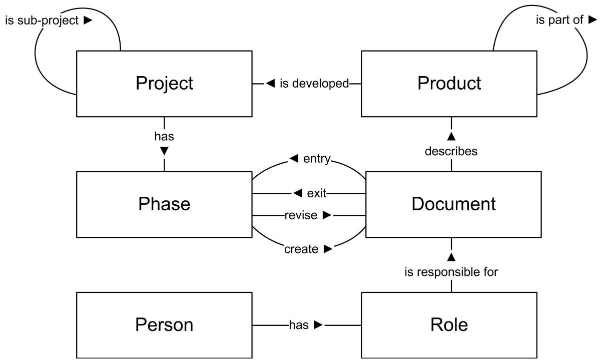


Abbildung 4: Beziehungen zwischen Artefaktebenen (vereinfachtes Schema).

Handler, der die Steuerung und das Delegieren der einzelnen Aufgaben übernimmt. Die einzelnen Unteranfragen werden an die jeweils zuständigen *Suchmodule* weitergeleitet. Bei reinen Attributfacetten ist dies die Facettierungskomponente, die die Anfragevorschauen für die einzelnen Facetten bestimmt. Bei Ähnlichkeitsfacetten kommen spezialisierte Module zum Einsatz, die die Ähnlichkeitsbestimmung übernehmen. Dies kann bspw. ein Modul für die 3D-Geometrieähnlichkeit oder eine Volltextsuche für textuelle Inhalte sein. Diese Module liefern jeweils ein Ranking zurück, das für die Achsendarstellung in den parallelen Koordinaten notwendig ist. Zusätzlich bestimmt die Facettierungskomponente noch die Linienzüge zur Repräsentation der einzelnen Artefakte in den ||-coords. Das *Rankingmodul* erstellt basierend auf den Einzelrankings der Unteranfragen sowie deren Präferenzfunktionen ein finales Ranking. Das Modul *Facettenauswahl* bestimmt auf Basis der aktuellen Anfrage und der Artefakttyphierarchien (inkl. der Facettenbeschreibungen) die Liste der aktuell wählbaren Facetten. Das Modul *Rechteprüfung* stellt sicher, dass der Nutzer nur Artefakte im Ranking zurückgeliefert bekommt, für die er mindestens Leserechte besitzt.

5.1 Beschreibung des Anfragemodells

Aufgrund der unterschiedlichen Visualisierungsmöglichkeiten und der verschiedenen Ausprägungen der Anfragearten ist es zunächst notwendig, auf die Charakteristika von Facetten genauer einzugehen. Grundsätzlich werden Attributfacetten und Ähnlichkeitsfacetten unterschieden. Attributfacetten stellen einen Aspekt eines Artefakts dar und können während der Indexierung bestimmt werden. Ähnlichkeitsfacetten hingegen werden zum Anfragezeitpunkt dynamisch ermittelt. Der Nutzer muss in diesem Fall eine Ähnlichkeitsanfrage starten (z. B. eine Schlagwortanfrage oder eine QbE-Anfrage mit einem Beispieldokument). Dieses Vorgehen wird in Abschnitt 5.3 detailliert.

Zusätzlich zu dieser Unterscheidung wird jeder Facette noch ein Facettentyp zugeordnet, der das Skalenniveau des Attributs beschreibt. Der LFRP-Ansatz unterscheidet *nominale*, *ordinale* und *metrische* Merkmale. Die Ausprägungen von nominalen und ordinalen Facetten sind unterscheidbar, wobei für letztere zusätzlich noch eine Rangfolge definiert ist. Wenn zusätzlich zur Rangfolge die Abstände zwischen den Facettenausprägungen bestimmbar sind, spricht man von metrischen Facetten.

Sowohl für die Indexierung als auch die Anfragestellung ist die Kardinalität von Facetten zu berücksichtigen. Facetten können *einwertig* oder *mehrwertig* sein, was sich auf die Kombinationsmöglichkeiten von Facettenwerten in der Anfrage auswirkt.

Eine Suchanfrage besteht aus einzelnen Unteranfragen für jede Facette, die durch eine Achse in den ||-coords dargestellt wird. Die verschiedenen Unteranfragen Q_i werden per Konjunktion (logisches UND) verknüpft. Die Verknüpfung der einzelnen Achsen mittels Disjunktion (logisches ODER) wird absichtlich nicht unterstützt, da diese erhöhte Komplexität der Anfrage das Verständnis für den Nutzer erschweren würde.

Die Anfragemöglichkeiten für die einzelnen Unteranfragen Q_i sind abhängig vom Facettentyp und von der Kardinalität einer Facette.

Bei einwertigen Facetten wird bei der Selektion von mehreren Facettenausprägungen automatisch die ODER-Verknüpfung verwendet, da die UND-Verknüpfung zu einer leeren Ergebnismenge führen würde. Eine einwertige Beispielfacette ist der *Dokumenttyp*, der für jedes Dokument eindeutig ist. Dokumente die gleichzeitig mehreren Dokumenttypen entsprechen existieren nicht.

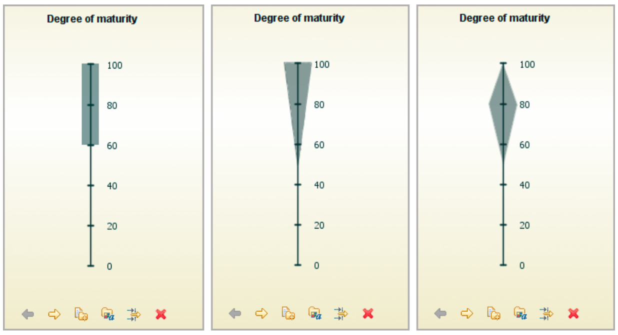


Abbildung 5: Beispiele für Präferenzfunktionen.

Bei mehrwertigen Facetten sind sowohl die Konjunktion als auch die Disjunktion anwendbar. Betrachtet man die mehrwertige Dokumentfacette *Autor* äußert sich die unterschiedliche Behandlung von Mehrfachselektionen. Bei Anwendung der Konjunktion werden nur die Dokumente zurückgeliefert, die von allen selektierten Autoren erstellt wurden. Die Disjunktion dagegen schwächt dieses Kritierium ab und liefert Dokumente, die von mindestens einem der selektierten Autoren erstellt wurden.

Die Selektion einzelner Werte bei metrischen Facetten ist möglich, allerdings wird die Filterung auf Intervallabschnitten der häufigere Anwendungsfall sein. Das Anfragemodell des LFRP-Anfrageframeworks unterstützt den Nutzer durch die Möglichkeit, mehrere Intervalle auf einer Facette anzugeben. Ein sinnvoller Einsatzzweck für diese Funktionalität ist bspw. der Vergleich von bestimmten Produkten mit verschiedenen Preisgruppen. Ein Beispiel für eine Intervallselektion findet sich in Abbildung 5 in der linken Achse. In diesem Beispiel werden nur die Dokumente zurückgeliefert, deren Reifegrad in einem Bereich zwischen 60% und 100% liegt.

5.2 Möglichkeiten zur Erstellung und Beeinflussung eines Rankings

Die klassische facettierte Suche ist eine Repräsentation des Bool'schen Retrievalmodells, also mengenbasiert. D. h. das Ergebnis besteht aus einer Menge von Artefakten, die keinerlei Rangfolge beinhalten. Da die Ergebnismengen im Unternehmensumfeld sehr groß werden können, ist es notwendig, dem Nutzer die Möglichkeit zur Verfügung zu stellen, Prioritäten für Facetten anzugeben, nach denen ein Ranking erfolgen soll. Der Nutzer kann über sogenannte *Präferenzfunktionen* festlegen, welche Facettenausprägungen bzw. Wertebereiche bei metrischen Facetten im Ranking höher gewichtet werden sollen. Mit Hilfe der nachfolgenden Formel kann eine Punktzahl $score(a_j)$ für jedes Artefakt a_j berechnet werden, nach der die Sortierung im Ranking vorgenommen wird.

$$score(a_j) = \sum_{i=1}^{n} \alpha_i \cdot f_i(x_{i,j}) \qquad (1)$$

$$\text{mit } \alpha_1 + \alpha_2 + \cdots + \alpha_n = 1$$

Hierbei beschreibt α_i die Gewichtung der Facette i, die der Nutzer über die dritte Schaltfläche von rechts in der Symbolleiste unter jeder Rankingfacettenachse einstellen kann. Abbildung 1 zeigt dies für die beiden äußersten Achsen. Der Nutzer kann damit eine niedrigere oder höhere

Gewichtung für einzelne Rankingkriterien festlegen. Standardmäßig beträgt die Gewichtung für jede Facette $1/n$, wobei n der Anzahl der gewählten Facetten entspricht. Die Funktion $f_i(x_{i,j}) \in [0,1]$ beschreibt die Nutzerpräferenzen für den Wert $x_{i,j}$ der Facette i für das Artefakt a_j und kann vom Nutzer graphisch definiert werden.

Der Nutzer kann die Gestaltung der Funktion f_i frei nach seinen Präferenzen vornehmen, d. h. das System gibt keine Restriktionen vor. Zum besseren und einfacheren Verständnis stellt das System jedoch ein paar einfache „Funktionsvorlagen“ zur Verfügung, die häufiger benötigt werden, wie bspw. die Dreiecks- und Vierecksfunktionen aus Abbildung 5. Diese Vorlagen können nach dem Hinzufügen durch direkte Manipulation weiter verfeinert und verändert werden. Jede Änderung der Funktion äußert sich in einer Neuberechnung des Rankings und der Facettenwerte.

5.3 Einbindung von QbE-Bedingungen

Zusätzlich zur Filterung von Artefakten benötigt der Produktentwickler die Möglichkeit mit Hilfe von Beispielartefakten nach ähnlichen Artefakten zu suchen. Das LFRP-Framework ermöglicht diese QbE-Suche durch die Bereitstellung von Ähnlichkeitsfacetten.

Häufig ist es für den Nutzer zielführend nach Schlagworten zu suchen. Bei diesen Volltextsuchen kommt häufig das Vektorraummodell zum Einsatz [Salton *et al.*, 1975]. Bei der Auswahl einer Ähnlichkeitsfacette wird initial eine Achse ohne Facettenbeschriftung als Platzhalter in die ||-coords eingefügt. Der Nutzer muss im zweiten Schritt die Beispielanfrage füllen, d. h. bei einer Textanfrage Schlagwörter übergeben. Dies geschieht über einen Eingabebereich unter der Achse (vgl. Achse *Text query* in Abb. 1).

Im Bereich der Produktentwicklung tritt häufig die Notwendigkeit auf, dass bereits vorhandene Komponenten wieder verwendet werden sollen, um so eine erneute kostspielige Entwicklung zu vermeiden. Um dies zu unterstützen bieten wir zusätzlich zu den textuellen Ähnlichkeitsfacetten QbE-Facetten an, die es dem Nutzer ermöglichen, einen Ähnlichkeitsvergleich von Produkten auf Basis der im Beispieldokument enthaltenen Geometrie oder auch der Topologie durchzuführen. Bei der Indexierung der Artefakte werden diese Informationen aus den zugehörigen Produktmodellen gewonnen (CAD → 3D, techn. Zeichnung → 2D). Hinsichtlich der Artefakttypen werden diese Informationen der Produktebene zugeordnet, um die produktorientierte Denkweise des Entwicklers zu unterstützen. Zur Repräsentation dieser Facetten wird auf bestehende Ansätze der Forschung zurückgegriffen (vgl. Kapitel 2).

Für den Nutzer arbeitet die Verwendung der Ähnlichkeitsfacetten analog zu den Attributfacetten. Dies ist von Vorteil, da der Nutzer die grundsätzliche Funktionsweise der Anfragesteuerung auf QbE-Anfragen übertragen kann und nur die Metapher der Facettenachse verstehen muss.

5.4 Dynamische Bereitstellung von Facetten

Im Enterprise Search-Umfeld findet sich eine Vielzahl von (semi-) strukturierten Artefakten, die mit Hilfe einer übergreifenden Suchfunktionalität zugänglich gemacht werden sollten. Da die Anzahl der beschreibenden Facetten aller Artefakte sehr groß werden kann, ist es notwendig, den Nutzer bei einer Informationssuche so zu unterstützen, dass ihm nur die jeweils aktuell sinnvollen und gültigen Facetten für eine weitere Verfeinerung der Suchanfrage angeboten werden. Der Inhalt der Liste der verfügbaren Facetten wird durch Auswertung der aktuellen Suchergebnisse und der Artefakttyphierarchien nach jeder Änderung der Suchanfrage aktualisiert.

Bei einer Anfrage auf der Produktebene, die der Nutzer über die Facette Produktgruppe „Schrauben“ einschränkt, werden zusätzlich zu den allgemeinen produktspezifischen Facetten noch Facetten, wie bspw. die Gewindeausführung, die Schraubenlänge und der Nenndurchmesser angeboten (vgl. Abbildung 3).

Die Bestimmung der aktuell gültigen Facetten wird vom Modul *Facettenauswahl* (siehe Abbildung 2) übernommen. Dies umfasst auch die Bestimmung der aktuell gültigen Verbindungen zu anderen Artefaktebenen.

5.5 Wechsel des Artefakttyps

Das LFRP-Suchframework unterstützt die Suche auf verschiedenen miteinander verbundenen Artefaktebenen. Die dazu notwendigen Beziehungen wurden in Abbildung 4 visualisiert. Eine Ebene wird durch einen einzelnen Artefakttyp definiert und enthält alle indexierten Artefakte dieses Typs. Die Beziehungen zwischen den verschiedenen Artefakttypen werden für den Ebenenübergang verwendet. Abbildung 6 zeigt beispielhaft Artefakte auf vier Ebenen. Auf der Produktebene sind Beziehungen zwischen Produkten sichtbar (bspw. *is-part-of* Beziehungen zwischen einzelnen Bauteilen und dem Gesamtprodukt) sowie Beziehungen zwischen Ebenen (Produkt *is-made-of* Material).

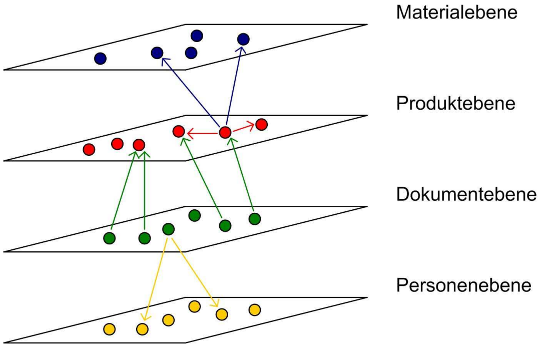


Abbildung 6: Beziehungen zwischen und innerhalb von Artefaktebenen.

Suchanfragen im LFRP-Suchframework können prinzipiell in jeder unterstützten Artefaktebene beginnen. Der Nutzer kann die Verlinkungen zwischen den Artefakten in zwei verschiedenen Anwendungsfällen nutzen.

Der erste Anwendungsfall besteht in der Suche nach Artefakten einer Ebene, indem der Nutzer seine Anfrage gemäß Abschnitt 5.1 formuliert. Um die Suchergebnismenge weiter zu verringern, kann es hilfreich sein, Filterungen mit Hilfe von Facetten, die von direkt verbundenen Ebenen stammen, vorzunehmen. Bei einer Suche auf der *Dokumentebene* kann es sinnvoll sein, die Ergebnisdokumente über die Produktfacette *Produktname* weiter einzuschränken. Gemäß Abbildung 4 können Dokumente Produkte beschreiben, d. h. es existiert eine Verbindung zwischen diesen beiden Artefakttypen, die es dem LFRP-Framework ermöglicht, Facetten der direkt verbundenen Ebenen auf Basis der aktuellen Ergebnismenge zu bestimmen und diese zur Filterung anzubieten. Das Anfrageframework generiert automatisch die dazu notwendige geschachtelte Anfrage. Dabei werden ausgehend von den aktuellen Artefakten des Suchergebnisses, in diesem Fall Dokumente, die damit verbundenen Produkte bestimmt. Basierend auf dieser Menge wird dann die Facette *Produktname* berechnet. Dem Nutzer wird damit die Möglichkeit gegeben, die Artefaktverlinkung zu nutzen, um weitere Filterkriterien zu definieren. Das Suchergebnis enthält nach wie vor nur Dokumente, die allerdings durch Facetten, die von direkt verbundenen Ebenen stammen, gefiltert werden können.

Der zweite Anwendungsfall beschreibt den Übergang zwischen Artefaktebenen. Analog zum ersten Anwendungsfall wählt der Nutzer Facetten zur Filterung des aktuellen Suchergebnisses aus und nimmt Selektionen darauf vor. Bei bestimmten Informationsbedürfnissen kann es sinnvoll sein, die Artefaktebene basierend auf den aktuellen Suchergebnissen zu wechseln. Ein Beispiel ist die Suche auf der Dokumentebene mit Hilfe eines Beispieldokuments. Der Nutzer kann die erhaltene Suchergebnismenge mit Dokumentfacetten weiter einschränken. Ist er mit den Ergebnisdokumenten zufrieden, kann der Wunsch aufkommen, alle durch die Dokumente beschriebenen Produkte zu sehen. Das Anfrageframework bestimmt jetzt basierend auf den Dokumenten in der Ergebnismenge die damit verbundenen Produkte. Das heißt, ausgehend von der aktuellen Ergebnismenge der Ursprungsebene navigiert das LFRP-Framework anhand der spezifizierten Beziehung zur Zielebene und erstellt die neue Ergebnismenge auf Basis der Artefakte in dieser Ebene. Das neue Suchergebnis kann weiter über Facettenselektionen eingeschränkt werden. Zusätzlich hat der Nutzer die Möglichkeit, zur Ursprungsebene zurückzugehen, um die initialien Filterkritierien zu verringern oder zu erweitern. Die Liste der aktuell gültigen Verlinkungen zu anderen Ebenen ist direkt neben der Liste der aktuell angebotenen Facetten zu finden (vgl. Abschnitt „Switch artifact types“ in Abbildung 1).

6 Zusammenfassung und Ausblick

In der vorliegenden Publikation haben wir das LFRP-Framework als einen interaktiven Ansatz zur Unterstützung komplexer Suchsituationen vorgestellt. Unser Ansatz kombiniert die etablierten Techniken der facettierten Suche, des Rankings und von parallelen Koordinaten in ein integriertes und mächtiges Werkzeug, um explorative Suchanfragen zu unterstützen. Dieser Ansatz wurde in einem Forschungsprojekt mit Industriebeteiligung aus der Domäne der technischen Produktentwicklung entwickelt. Erste Nutzerbefragungen zeigen, dass die grundsätzliche Ausrichtung des Ansatzes erfolgversprechend ist.

Nichtsdestotrotz existieren weitere offene Punkte, die noch zu untersuchen sind. Wir entwickeln ein Konzept, um verschiedene Nutzergruppen zu unterscheiden. Expertennutzer sollen hierbei auf die vollständige Funktionalität des Ansatzes zurückgreifen können, um Suchvorlagen für verschiedene Situationen im PEP, die durch komplexe Informationsbedürfnisse geprägt sind, zu definieren. Diese Vorlagen werden in eine Prozessmanagementlösung integriert und können von Standardnutzern als Einstiegspunkt für interaktive Suchanfragen genutzt werden.

Ein weiterer Aspekt betrifft die Präsentation der Ergebnisliste. Hierbei sind weitere Visualisierungen vorstellbar, bspw. Schlagwortwolken (Tag clouds, vgl. [Hassan-Montero und Herrero-Solana, 2006]) als Repräsentation für interessante Aspekte wie Projektwolken oder Autorwolken bzw. erweiterte Vorschaufunktionalitäten für die Artefakte.

Ferner müssen auch effiziente Implementierungen des Frameworks und die sinnvolle Verteilung der Berechnungen zwischen Frontent und Backend betrachtet werden.

Danksagung

Diese Arbeit entstand im Rahmen des von der Bayerischen Forschungsstiftung geförderten Forschungsverbundes FORFLOW (*http://www.forflow.org*).

Literatur

[Eckstein und Henrich, 2008] Raiko Eckstein und Andreas Henrich. Reaching the Boundaries of Context-Aware IR: Accepting Help from the User. In *Proc. of the 2nd Int. Workshop on Adaptive Information Retrieval (AIR 2008)*, London, 2008. BCS.

[Eckstein und Henrich, 2009] Raiko Eckstein und Andreas Henrich. Visual Browsing in Product Development Processes. In The Design Society, *Proc. of the 17th Int. Conf. on Engineering Design*, 2009. (angenommen)

[Grabowski und Geiger, 1997] H. Grabowski und K. Geiger, Herausgeber. *Neue Wege zur Produktentwicklung*. Dr. Josef Raabe Verlags-GmbH, Stuttgart, 1997.

[Hassan-Montero und Herrero-Solana, 2006] Y. Hassan-Montero und V. Herrero-Solana. Improving Tag-Clouds as Visual Information Retrieval Interfaces. In *Int. Conf. on Multidisciplinary Information Sciences and Technologies*, 2006.

[Inselberg, 1985] A. Inselberg. The Plane with Parallel Coordinates. *The Visual Computer*, 1(4):69–91, 1985.

[Karnik *et al.*, 2005] M. V. Karnik, S. K. Gupta, D. K. Anand, F. J. Valenta, I. A. Wexler. Design Navigator System: A Case Study in Improving Product Development through Improved Information Management. In *ASME Computers and Information in Engineering Conference*, Long Beach, CA, 2005.

[Marchionini, 2006] Gary Marchionini. Exploratory Search: from Finding to Understanding. *Communications of the ACM*, 49(4):41–46, 2006.

[Osada *et al.*, 2002] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin. Shape Distributions. *ACM Transactions on Graphics*, 21:807–832, 2002.

[Reitz, 2004] Joan M. Reitz. *Dictionary for Library and Information Science*. Libraries Unlimited, Westport, Conn., 2004.

[Salton *et al.*, 1975] Gerard Salton, Anita Wong, und Chung-shu Yang. A Vector Space Model for Automatic Index. *Comm. of the ACM*, 18(11):613–620, 1975.

[Schichtel, 2002] Markus Schichtel. *Produktdatenmodellierung in der Praxis*. Carl Hanser Verlag, 2002.

[Sundar *et al.*, 2003] H. Sundar, D. Silver, N. Gagvani, S. Dickinson. Skeleton Based Shape Matching and Retrieval. In *Proc. of the Int. Conf. on Shape Modeling and Applications*, S. 130–139, 2003. IEEE Comp. Society.

[Vranic, 2004] D. V. Vranic. *3D Model Retrieval*. Dissertation. Universität Leipzig, 2004.

[Weber und Henrich, 2007] N. Weber und A. Henrich. Retrieval of Technical Drawings in DXF Format - Concepts and Problems. In *Lernen - Wissen - Adaption, Workshop Proc.*, S. 213–220, 2007. Universität Halle-Wittenberg.

[Yee *et al.*, 2003] K.-P. Yee, K. Swearingen, K. Li, M. Hearst. Faceted Metadata for Image Search and Browsing. In *Proc. of SIGCHI Conf. on Human Factors in Computing Systems*, S. 401–408, New York, 2003. ACM Press.

Visual Resampling for Pseudo-Relevance Feedback during Speech-based Video Retrieval

Stevan Rudinac, Martha Larson, Alan Hanjalic
Delft University of Technology, Faculty of EEMCS, Department of Mediamatics/ ICT Group
Mekelweg 4, 2628 CD Delft, The Netherlands
{s.rudinac, m.a.larson, a.hanjalic}@tudelft.nl

Abstract

A method is proposed that makes use of visual reranking to selectively sample feedback sets for Pseudo-Relevance-Feedback during speech-transcript-based video retrieval. Observed performance improvement is indicative of the ability of visual reranking to increase the relevance density of the feedback set.

1 Introduction

Applying resampling techniques to select feedback documents has been shown to improve performance of pseudo-relevance feedback (PRF) [Lee *et al.*, 2008]. The success of reranking techniques for video retrieval, e.g., [Hsu *et al.*, 2006] and [Yan *et al.*, 2003], suggests that judicious application of visual information can refine results lists generated by text-based retrieval. We propose an approach to speech-transcript-based video retrieval, which we designate Visual Reranking+PRF (VR+PRF), that combines these two principles. VR+PRF informs feedback document resampling with document similarity patterns derived from low level visual features. In conventional PRF, the top N items in the initial results list are used as the feedback set. VR+PRF also samples the top N items, but from a refined version of the results list created by the application of visual reranking. In other words, our method succeeds in making use of "visually" top ranked videos as positive feedback documents for PRF. In this respect, it is different from the existing techniques applying PRF in video retrieval. These techniques typically do not perform reranking and exploit only textual information from the top-ranked documents or the visual information from lower-ranked ones ("visual" pseudo-negatives) [Yan *et al.*, 2003].

We show that VR+PRF achieves a modest but consistent improvement in retrieval performance when compared to conventional PRF. This performance gain reflects improved relevance density resulting from visual reranking. Further analysis demonstrates that visual resampling is particularly helpful for certain queries, and that its positive impact cannot easily be matched by text-based resampling.

2 Visual Reranking

Our use of visual features is based on the expectation that the process of video production is constrained by conventions that give rise to visual similarity among videos that treat the same topic. If visual similarity among items in the results list can be assumed to result from topical relation to the query, then sampling items from dominant visual clusters should yield a high precision feedback set with potential to improve PRF. We build on our previous work that introduced the use of the Average Item Distance (AID) for the visual reranking of results lists [Rudinac *et al.*, 2009]. Documents d_i in the results list R are reranked by their *AID* score, which is their average distance to all other documents in the results list, expressed by

$$AID_{d_i} = \frac{\sum_{d_j \in R : d_j \neq d_i} dist(d_j, d_i)}{|R| - 1} \qquad (1)$$

The AID score is similar to the aggregate dissimilarity used in [Yan *et al.*, 2003]. The effect of the AID score is to advantage documents that are sufficiently representative of the entire collection. This could be the documents that are either central to the overall set of documents, or that belong to the most dominant clusters occurring in the collection. Resampling documents with top AID scores is used with a similar motivation and to a similar end as the cluster-based resampling used in [Lee *et al.*, 2008].

3 Experimental Setup

For our experiments we use the VideoCLEF 2008 test set [Larson *et al.*, 2008], which contains Dutch-language documentaries automatically segmented into shots. In the data set, each shot is associated with an automatically extracted keyframe and a speech recognition transcript of its contents. To make our experiments independent of the challenges that are beyond the scope of this paper, we discard the shots without speech transcripts, arriving at a test collection containing a total of 5,362 shot-level documents. The queries used (total of 53) are semantic labels that have been hand-assigned by archivists and are used at the video archive for annotating and retrieving video. Video retrieval and reranking are performed in an automatic manner. We carry out retrieval using the Lemur toolkit, choosing the Kullback-Leibler divergence model [Lafferty *et al.*, 2001] with Dirichlet smoothing. PRF involves a linear combination of the original query model and the feedback model. Visual reranking consists of reordering documents by increasing AID score, where *dist* is the Euclidean distance between vectors of low-level visual features including color moments and Gabor texture features extracted from the keyframe representing each shot, described in [Rudinac *et al.*, 2009].

4 Experimental Results

4.1 PRF vs. VR+PRF

Our first experiment compares conventional PRF and VR+PRF, testing application of the method in a single and then repeated iterations, up to the point where performance gains cease. Performance (cf. Table 1) is reported as the Mean Average Precision (MAP) at optimal parameter settings for each round.

Condition	1x	2x	3x	4x
PRF	0.2161	0.2523	0.2573	0.2586
VR+PRF	0.2191	0.2661	0.2667	0.2678

Table 1: Performance (MAP) of PRF and VR+PRF iterations (Parameters optimized individually for each condition)

VR+PRF demonstrates a small but consistent improvement over conventional PRF at each round. To investigate the source of improvement, we follow [Lee *et al.*, 2008] and calculate the relevance density for various sizes of feedback sets (cf. Table 2). The relevance density for feedback set of size N is the proportion of documents in the set that are relevant. Recall that our feedback set is the top N documents in the retrieved list for PRF and in the visually-reranked retrieved list for VR+PRF.

Condition	5	10	15	20	30	100
1x-PRF	0.419	0.415	0.418	0.420	0.431	0.435
2x-PRF	0.468	0.470	0.473	0.475	0.478	0.460
3x-PRF	0.476	0.479	0.484	0.489	0.494	0.473
4x-PRF	0.479	0.483	0.489	0.494	0.496	0.478
1x-VR+PRF	0.434	0.438	0.429	0.429	0.438	0.455
2x-VR+PRF	0.513	0.493	0.496	0.496	0.499	0.492
3x-VR+PRF	0.509	0.504	0.503	0.504	0.505	0.489
4x-VR+PRF	0.513	0.496	0.502	0.502	0.502	0.488

Table 2: Relevance density of feedback sets of increasing size

Note that for smaller feedback sets, iterations of VR+PRF yield higher relevance densities. Additionally, a relevance density peak with a feedback set of size 30 cuts across conditions.

5 VR+PRF vs. Textual Reranking+PRF

Our second experiment compares VR+PRF with Textual Reranking+PRF (TR+PRF). Textual reranking is accomplished by reordering items in the result list by their *AID* scores, calculated using distances in the text space. For this exploratory experiment we use feedback set of size 30, revealed previously to be a relevance density peak. Other parameters were set to reasonable global optima we judged would not advantage a particular test condition. The results in Table 3 demonstrate that the improvement of VR+PRF cannot be easily matched by a simplistic form of text-based feedback document resampling.

Condition	1x	2x	3x	4x
PRF	0.2032	0.2263	0.2262	0.2262
VR+PRF	0.2031	0.2323	0.2333	0.2181
TR+PRF	0.2030	0.2153	0.2086	0.2011

Table 3: Performance (MAP) of PRF, VR+PRF and TR+PRF iterations

A visualization of the performance of queries (for clarity limited to those that achieve MAPs > 0.10) reveals that the strength of VR+PRF is based on its power to improve certain queries. TR+PRF, on the other hand tends to yield the same level of performance as conventional PRF, or else hurt queries.

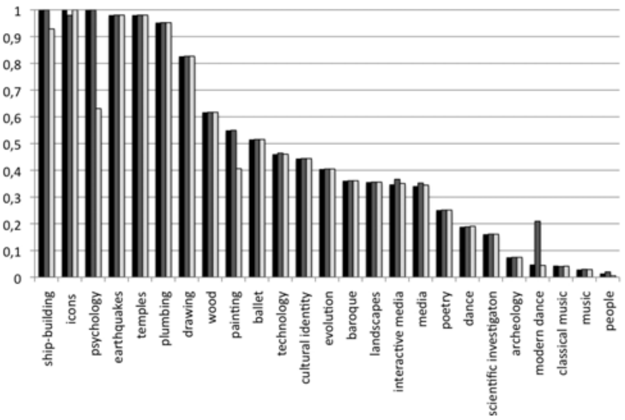


Figure 1: Query by query average precision of 2xPRF (dark), 2xVR+PRF (middle), and 2xTR+PRF (light)

6 Outlook

We have proposed the use of visual reranking to support the selective sampling of a feedback set for use in PRF feedback during speech-based video retrieval. Future work will involve investigation of other visual reranking methods and exploration of query classification methods that will allow us to predict which queries stand to benefit from the application of VR+PRF.

Acknowledgments

The research leading to these results has received funding from the European Commission's Seventh Framework Programme under grant agreement n° 216444 (PetaMedia Network of Excellence).

References

[Rudinac *et al.*, 2009] S. Rudinac, M. Larson, and A. Hanjalic. Exploiting Visual Reranking to Improve Pseudo-Relevance Feedback for Spoken-Content-Based Video Retrieval. In *Proc. WIAMIS'09*.

[Hsu *et al.*, 2006] W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Video search reranking via information bottleneck principle. In *Proc. ACM MM'06*.

[Lafferty *et al.*, 2001] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proc. SIGIR'01*.

[Larson *et al.*, 2008] M. Larson, E. Newman, and G. Jones. Overview of VideoCLEF 2008: Automatic generation of topic-based feeds for dual language audio-visual content. In *CLEF 2008 Workshop*.

[Lee *et al.*, 2008] K. S. Lee, W. B. Croft, and J. Allan. A cluster-based resampling method for pseudo-relevance feedback. In *Proc. SIGIR'08*.

[Tian *et al.*, 2008] X. Tian, L. Yang, J. Wang, Y. Yang, X. Wu, and X.-S. Hua. Bayesian video search reranking. In *Proc. ACM MM'08*.

[Yan *et al.*, 2003] R. Yan, A. G. Hauptmann, and R. Jin. Negative pseudo-relevance feedback in content-based video retrieval. In *Proc. ACM MM'03*.

Inhaltsbasierte Suche in audiovisuellen Medien auf Basis von Texterkennung

Stephan Heinich, Jens Kürsten
Technische Universität Chemnitz
09111, Chemnitz, Deutschland
[vorname.nachname]@informatik.tu-chemnitz.de

Zusammenfassung

Der vorliegende Artikel gibt einen Überblick über ein entwickeltes Textextraktionssystem und die Weiterverwendung der Ergebnisse in einer inhaltsbasierten Video-Suche. Dazu wird das bestehende Retrieval-System für die Verarbeitung der teilweise ungenauen Ergebnisse der OCR erweitert. Das Demo-System umfasst die Beschreibung der Teilprozesse der Texterkennung sowie ein prototypisches Retrieval Interface, welches über Webservice-Schnittstellen an das Recherche Framework Xtrieval angebunden ist.

1 Motivation

Im InnoProfile Projekt sachsMedia[1] werden Verfahren zur inhaltlichen Analyse von audiovisuellen Medien erforscht. Diese Methoden werden benötigt um beispielsweise historisch gewachsene Archive von lokalen Fernsehsendern zugänglich und damit wirtschaftlich verwertbar zu machen.

Der nachfolgend beschriebene Prototyp umfasst ein System zur Texterkennung im Videomaterial, eine Webservice-Architektur für die Indizierung und Suche in einem Beispielkorpus und Konzepte zur nachträglichen Korrektur der OCR Resultate. Der Testkorpus umfasst ca. 400 Stunden sendefähiger Fernsehbeiträge mit intellektuell vergebenen Metadaten. Zur Recherche sind folgende Deskriptoren der Beiträge verwertbar: Titel (für 100%), Beschreibungen (für ca. 70%) und Stichworte (für ca. 45%). Zur inhaltsbasierten Suche wird der Index mit den Ergebnissen der Texterkennung angereichert. Die Extraktion der Texte und die automatische Korrektur der Resultate sind dabei die größte Herausforderung.

Der beschriebene Prototyp soll in Zukunft zur inhaltlichen Erschließung von Archiven genutzt werden, die wenig oder überhaupt keine intellektuellen Annotationen enthalten.

2 Verwandte Arbeiten

Ein System zur Recherche in verrauschten textuellen Daten, die mit automatischen Verfahren extrahiert wurden, sollte die inhaltlichen Fehler der Erkennung im Idealfall eliminieren oder zumindest weitestgehend verringern. Nach [Beitzel et. al., 2003] können Ansätze zur Lösung des Problems in folgende Kategorien unterteilt werden: (a) spezielle IR Modelle für OCR Korpora [Harding et. al., 1997], [Taghva et. al., 1994], (b) automatische Korrektur der Fehler bei der automatischen Erkennung [Liu et. al. 1991], [Tong et. al., 1996] und (c) Verbesserung des String-basierten Ähnlichkeitsmaßes für verrauschten Daten [Collins-Thompson et. al., 2001], [Brill & Moore, 2000].

In mehreren zentral organisierten Kampagnen, wie dem TREC-5 Confusion Track[2] oder CLEF CL-SR[3] wurden verrauschte Korpora zur Evaluation von Retrieval-Systemen erstellt und eingesetzte Verfahren verglichen.

Die durchgeführten Studien zeigen, dass unter den Voraussetzungen einer geringen Fehlerrate bei der Erkennung und ausreichend langen Dokumenten (bspw. gescannte Buchseiten oder Transkriptionen von Tondokumenten) die Genauigkeit der Recherche mit den verwendeten Ansätzen nur unwesentlich verschlechtert wird. Jedoch sind genau diese Bedingungen hier im konkreten Problemfeld nicht gegeben. Einerseits erschwert die hohe Varianz der Eigenschaften des auftretenden Textes im Videomaterial die automatische Erkennung und andererseits sind die auftretenden Textpassagen generell eher kurz.

Im folgenden Abschnitt wird das Verfahren zur Erkennung und Extraktion von Text beschrieben. In Abschnitt 4 werden die zwei verwendeten Ansätze zur Indizierung der OCR-Ergebnisse erläutert.

3 Aufbau der Textextraktion

Analog zu [Gargi et. al., 1998] besteht das nachfolgend beschriebene prototypische Textextraktionssystem aus vier Hauptschritten. Dabei werden im ersten Schritt die Textkandidaten (Einzelframes) im Video detektiert. Anschließend folgen Textlokalisierung und Segmentierung von Vorder- und Hintergrund. Daraufhin werden die Textbereiche verbessert und binarisiert. Im letzten Schritt wird der so aufbereitete Frame einem OCR-System zugeführt um den extrahierten Text zu erkennen. Das vollständige Verfahren wird in [Heinich, 2009] umfassend erläutert.

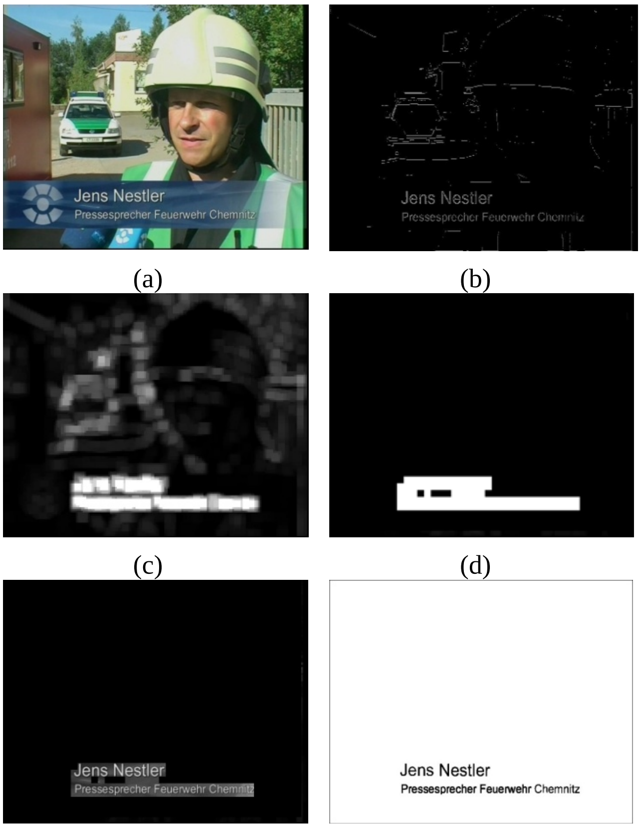
Aufgrund der Redundanz in aufeinanderfolgenden Frames und dem Fakt das Text mindestens zwei Sekunden dargestellt werden muss um gelesen werden zu können, ist es möglich nur jeden zehnten oder zwanzigsten Frame zu verarbeiten.

Zur Veranschaulichung soll Abbildung 1 dienen, in der die einzelnen Schritte der Prozesskette anhand eines Beispiels dargestellt sind.

[1] Gefördert durch: Unternehmen Region, Die BMBF Innovationsinitiative Neue Länder

[2] http://trec.nist.gov/data/t5_confusion.html

[3] http://www.clef-campaign.org/2005/working_notes/CLEF2005WN-Contents1.htm#Clsr



(a) (b) (c) (d) (e) (f)

Abbildung 1: (a) Original-Frame, (b) Bild der relevanten Kantenpixel, (c) Gewichtungsmaske der DCT, (d) normalisierte Maske, (e) Maskierungsergebnis, (f) Resultat für OCR-API

3.1 Textdetektion

In diesem Prozessschritt werden Frames bestimmt, die Text enthalten können. Dabei wird mit Hilfe einer Heuristik überprüft, ob sich genügend Kantenpixel mit einem Mindestgradienten im aktuellen Frame befinden. Dazu werden zwei Schwellwerte verwendet. Ersterer dient zur Bestimmung von Kantenpixeln im Bild und ein Zweiter beschreibt die Mindestanzahl solcher Kantenpixel in einem Textkandidaten. Die Abbildung 1b repräsentiert die relevanten Kantenpixel im Frame.

3.2 Textlokalisierung und -segmentierung

Nach [Lu & Barner, 2008] wird zur Textlokalisierung das Verfahren der gewichteten diskreten Cosinus-Transformation (DCT) verwendet. Dabei wird mit Hilfe der DCT das Spektrum eines Makroblocks ermittelt. Im hier vorgestellten Verfahren wird die DCT auf 16x16 Pixel große Makroblöcke angewendet. Mögliche Textregionen befinden sich im mittleren Frequenzbereich und können mit folgender Gewichtungsvorschrift verstärkt werden:

$$W_e(p,q) = \begin{cases} 0, & p+q<4 \\ C(p,q)^2, & 4 \le p+q < 12 \\ 0, & p+q \ge 12 \end{cases}$$

Zusätzlich werden damit die nieder- und hochfrequenten Anteile des Spektrums eliminiert. Hierbei beschreiben p und q die Koordinaten der Pixel in einem Makroblock. Die gewichteten Koeffizienten $W_e(p,q)$ resultieren aus dem Quadrat des Koeffizienten $C(p,q)$ um irrelevante Frequenzen abzuschwächen. Die Energie E eines Textblockes wird letztlich mit

$$E = \sum_{p=0}^{N-1} \sum_{q=0}^{N-1} W(p,q)$$

bestimmt. Nach einer Normalisierung der Energiewerte ergibt sich eine Maske wie sie in Abbildung 1c dargestellt ist. Diese Maske wird mit einem simplen Schwellwertverfahren binarisiert. Zusammenhängende Textblöcke werden durch umschließende Rechtecke begrenzt. Textblöcke mit einer zu geringen Größe und ungeeignetem Höhen-Breitenverhältnis werden eliminiert. Dies ist in Abbildung 1d illustriert. Im nächsten Schritt wird der Frame wie in Abbildung 1e maskiert.

3.3 Textbinarisierung

Bei der Binarisierung geht es darum, den Frame in zwei Farben zu unterteilen. Um ein optimales Eingangsbild für die OCR-API zu erreichen, wird der Text schwarz gefärbt und der Hintergrund weiß. Dazu wird ein Regionenbasiertes Schwellwertverfahren auf die umschließenden Rechtecke angewendet. Text- und Hintergrundfarbe werden ermittelt und ein automatischer Schwellwert für die Binarisierung festgelegt.

Die Annahme bei der Bestimmung von Text und Hintergrund besteht darin, dass in einem umschließenden Rechteck weniger Textpixel enthalten sind als Hintergrundpixel. Gibt es in dem Histogramm eines Rechtecks zwei große Peaks so ist der größere die Hintergrundfarbe und der kleiner die Textfarbe. Sollte es sich nur um einen signifikanten Peak handeln wird angenommen, dass der Hintergrund keine einheitliche Farbe besitzt und somit der Peak die Textfarbe beschreibt. Das Ergebnis dieser Binarisierung (Abbildung 1f) wird dann der Texterkennung übergeben.

3.4 Texterkennung

Zur Texterkennung wurden ein kommerzielles SDK (Asprise OCR[4]), zwei Consumer-Produkte (Omnipage 16 Professional[5], Finereader[6]) und ein Open-Source-Projekt (tessereact-ocr[7]) verglichen. Die Asprise-API lieferte im Vergleich die schlechtesten Ergebnisse. Omnipage und Finereader lieferten bessere Resultate, besitzen aber ausschließlich ein grafisches Nutzer-Interface und keine Programmierschnittstelle und lassen daher keine Einbindung in einen automatisierten Prozess zu.

Die OCR-API tesseract erzielte im Vergleich die besten Erkennungsraten und wird daher in der hier beschriebenen Implementierung eingesetzt. Das Projekt ging aus einem HP-Labs-Forschungsprojekt hervor und wurde 2007 in einem Google Code Projekt übernommen.

[4] http://asprise.com/product/ocr/index.php?lang=java
[5] http://www.nuance.de/
[6] http://finereader.abbyy.de/
[7] http://code.google.com/p/tesseract-ocr/

4 Anbindung an ein Retrieval-System

Über eine Webservice-Architektur [Kürsten, 2009] werden sämtliche Resultate der automatischen Verfahren der Inhaltsanalyse an das Retrieval-System Xtrieval [Wilhelm, 2008] übergeben. Für die Recherche im Testkorpus wird Lucene[8] mit einer Variante des Vektorraummodells verwendet. Vor der Indizierung werden die Ergebnisse des beschriebenen OCR Verfahrens aufbereitet. Dies ist notwendig, weil die Erkennung auf dem Ausgangsmaterial einerseits im hohen Maße redundante Ergebnisse produziert und andererseits Teile des erkannten Textes fehlerhaft sind. Diesen Problemen wird mit den beiden nachfolgend beschriebenen Verfahren entgegnet. Eine wissenschaftliche Evaluation beider Ansätze im vorliegenden Anwendungsfall steht noch aus.

4.1 Automatische Korrektur mit Wörterbuch

Bei der automatischen Erkennung kommt es häufiger zu einfachen Fehlern. Diese seien folgendermaßen definiert: Ein Wort hat eine kleine Buchstabenfehlerrate, wenn das Verhältnis richtig erkannter Buchstaben zur Gesamtanzahl erkannter Buchstaben größer als 0.8 ist.

Zur Eliminierung der Fehler wird die frei verfügbare API Suggester Spellcheck[9] eingesetzt. Diese liefert für einen übergebenen Ausdruck mehrere Vorschläge zurück, aus denen ein Benutzer das letztlich korrekte Wort auswählen kann. In der automatischen Verarbeitung ist diese Selektion nicht realisierbar. Ausgehend von der Annahme, dass ein nicht vollständig erkanntes Wort eine kleine Buchstabenfehlerrate aufweist, werden die ersten drei Vorschläge des Korrekturverfahrens an die Indizierung übergeben. Dieser Ansatz eliminiert einen großen Teil der redundanten und fehlerbehafteten Eingaben, bei denen unterschiedliche Buchstaben eines mehrfach erkannten Ausdrucks falsch erkannt werden.

Die verwendete Korrektur-API nutzt einen proprietären Algorithmus, der auf dem Abstand von Zeichen bei Tastatureingaben basiert und kombiniert diesen mit einem Sprachspezifischen Wörterbuch. Damit lässt sich die Redundanz der erkannten Terme pro Beitrag bereits erheblich reduzieren. Die Gesamtzahl der OCR-Terme verringert sich damit von 9993 auf 2534, was einer durchschnittlichen Dokumentlänge von ca. 26 Termen entspricht. Dennoch scheint es sinnvoll das Korrekturverfahren analog zu dem Ansatz in [Liu et. al., 1991], [Brill & Moore, 2000] oder [Höhn, 2008] zu erweitern. Dazu wird der Korrektur-Algorithmus an die spezifischen, häufig auftretenden Fehler der OCR (bspw. "nr" - "m", "ni" - "m", usw.) angepasst.

4.2 Dekomposition im Indizierungsprozess

Zur Anreicherung der vorhandenen intellektuellen Metadaten der Videos oder der automatischen Erschließung von Archivmaterial werden die OCR-Ergebnisse indiziert. Als alternativer Ansatz zur automatischen Korrektur wird eine spezielle Variante [Wagner, 2005] der N-Gram Tokenisierung [De Heer, 1974] verwendet. Dabei werden die Terme in einzelne Silben zerlegt und der eigentliche Index auf deren Basis erstellt. Das Ranking der Dokumente erfolgt anhand der Silben - eine Suchanfrage wird also mit den Silben jedes Terms verglichen. Sollte dabei ein von der OCR-Komponente erzeugter Term Fehler beinhalten, kann er bei einer Suche dennoch gefunden werden, wenn mindestens eines der erzeugten N-Gramme noch korrekt ist.

Das Dekompositionsverfahren nach [Wagner, 2005] wurde bereits mehrfach in umfangreichen Evaluationen[10, 11] mit dem Stemmer nach Porter[12] verglichen. Dabei zeigte sich, dass das Verfahren in Bezug auf die durchschnittliche Genauigkeit (MAP) signifikant bessere Ergebnisse als eine einfache Wortstamm-Reduktion erzielt. Nachteile des N-Gram Ansatzes im Retrieval sind jedoch die deutlich höhere Antwortzeit, die durch das Matching auf Silbenebene entsteht und der deutlich größere Speicherplatzbedarf für den invertierten Index der Silben [McNamee & Mayfield, 2004].

5 Zusammenfassung und Ausblick

Im vorliegenden Artikel wurde ein prototypisches System zur automatischen Erkennung von Text in Fernsehbeiträgen zur automatischen Erschließung von geringfügig oder nicht annotierten Archiven beschrieben. Erste Ergebnisse der Indizierung der OCR Resultate zeigen Potential für eine weitere Optimierung des Gesamtsystems.

Ein grundlegender Schritt hierzu wäre der Vergleich der beiden Verfahren anhand eines Evaluationskorpus. Ferner sollte das verwendete automatische Korrekturverfahren mit einem ähnlichen Ansatz[13] verglichen werden, um dessen Robustheit zu verifizieren. Weiterhin könnte ein adaptives Verfahren zur automatischen Erkennung und Korrektur von Fehlern eingesetzt werden. Abschließend sollte versucht werden das automatische Korrekturverfahren mit der N-Gram Tokenisierung zu kombinieren und mit den Ergebnissen der jeweils einzelnen Ansätze zu vergleichen.

Literatur

[Beitzel et. al., 2003] Beitzel, S. M., Jensen, E. C., Grossman, D. A. *Retrieving OCR Text: A Survey of Current Approaches.* Symposium on Document Image Understanding Technologies, Greenbelt, USA, 2003.

[Brill & Moore, 2000] Brill, E., Moore, R. C. *Improved string matching under noisy channel conditions.* Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, Seiten: 286-293, Hong Kong, 2000.

[Collins-Thompson et. al., 2001] Collins-Thompson, K., Schweizer, C., Dumais, S. *Improved String Matching Under Noisy Channel Conditions.* Proceedings of the 10th International Conference on Information and Knowledge Management, Seiten: 357-364, Atlanta, USA, 2001.

[8] http://lucene.apache.org/

[9] http://softcorporation.com/products/spellcheck/

[10] http://www.clef-campaign.org/2006/working_notes/workingnotes2006/kuerstenCLEF2006.pdf

[11] http://www.clef-campaign.org/2008/working_notes/kuersten-ah-paperCLEF2008.pdf

[12] http://snowball.tartarus.org/texts/germanic.html

[13] http://today.java.net/pub/a/today/2005/08/09/didyoumean.html?page=3

[De Heer, 1974] De Heer, T. *Experiments with Syntactic Traces in Information Retrieval.* Information Storage and Retrieval, Vol. 10, Seiten: 133-144, 1974.

[Gargi et. al., 1998] Gargi, U., Antani, S., Katsuri, R. *Indexing Text Events in Digital Video Databases.* Proceedings of the 14th Int. Conf. on Pattern Recognition, Seiten: 1481-1483, Brisbane, Australia, 1998.

[Harding et. al., 1997] Harding, S. M., Croft, W. B., Weir, C. *Probabilstic Retrieval of OCR Degraded Text Using N-Grams.* Proceedings of the 1st European Conference on Digital Libraries, Seiten: 345-359, Pisa, Italy, 1997.

[Heinich, 2009] Heinich, Stephan. *Textdetektion und -extraktion mit gewichteter DCT und mehrwertiger Bildzerlegung.* Workshop Audiovisuelle Medien, Seiten: 151-162, Chemnitz, Germany, 2009.

[Höhn, 2008] Höhn, Winfried. *Heuristiken zum Postprocessing von OCR-Ergebnissen.* Workshop LWA 2008 - FGIR, Seiten: 78-82, Würzburg, Germany, 2008.

[Kürsten, 2009] Kürsten, Jens. *Entwurf einer Service-orientierten Architektur als Erweiterung einer Plattform zum Programm-Austausch.* Workshop Audiovisuelle Medien, Seiten: 185-193, Chemnitz, Germany, 2009.

[Liu et. al., 1991] Liu, L.-M., Yair, M. B., Wei, S., Chan K.-K. *Adaptive post-processing of OCR text via knowledge acquisition.* Proceedings of the 19th annual conference on Computer Science, Seiten: 558-569, San Antonio, USA, 1991.

[Lu & Barner, 2008] Lu, S., Barner, K. E. *Weighted DCT based Text Detection.* Acoustics, Speech and Signal Processing, Seiten: 1341-1344, Las Vegas, USA, 2008.

[McNamee & Mayfield, 2004] McNamee, P., Mayfield, J. *Character N-Gram Tokenization for European Language Text Retrieval.* Information Retrieval, Vol. 7, Seiten: 73-97, 2004.

[Taghva et. al., 1994] Taghva, K., Borsack, J., Condit, A. *Results of Applying Probabilstic IR to OCR Text.* Proceedings of the 17th International Conference on Research and Development of Information Retrieval, Seiten: 202-211, Dublin, Ireland, 1994.

[Tong et. al., 1996] Tong, X., Zhai, C., Milic-Frayling, N., Evans, D. A. *OCR Correction and Query Expansion for Retrieval on OCR Data – CLARIT TREC-5 Confusion Track Report.* 5th Text Retrieval Conference (TREC-5), Seiten: 341-346, Gaithersburg, USA, 1996.

[Wagner, 2005] Wagner, Stefan. *A German Decompounder*, TU Chemnitz, Seminararbeit, 2005.

[Wilhelm, 2008] Wilhelm, Thomas. *Entwurf und Implementierung eines Frameworks zur Analyse und Evaluation von Verfahren im Information Retrieval.* TU Chemnitz, Fakultät für Informatik, Diplomarbeit, 2008.

Visualizing steps for shot detection

Marc Ritter and Maximilian Eibl
Chemnitz University of Technology
D-09111, Chemnitz, Germany
{ritm,eibl}@cs.tu-chemnitz.de

Abstract

This article introduces the current research and teaching framework, developed at the Chair Media Informatics at Chemnitz University of Technology. Within the demo-session we demonstrate and visualize its functionality on a scientific method for shot detection.

1 Introduction

The field of documentation of steadily increasing amounts of data and digital archiving is one of the challenging topics in the investigations of current research.

The *Retrieval Group* from the research project *sachsMedia — Cooperative Producing, Storage, Retrieval and Distribution of Audiovisual Media* is currently engaged in the extraction and indexing of important informations of predefined objects from the video sources of local television stations in preparation for successive and custom-driven search processes. [sachsMedia, 2009]

The close relationship of the project specific fields speech analysis (SPR), video analysis (VID), and meta-data handling (MDH) led to the development of the common framework *AMOPA*, which is described in more detail in section 2.

The field of methods proposed for object detection in literature is huge. Applied onto videos, not a small part of this algorithmic class fails due to abrupt changes of content within consecutive frames (*scene change*). The detection of shot boundaries is also widely discussed and became a major step in preprocessing, usually used to minimize the failure in postponed object detection. We are using an approach from [Liu *et al.*, 2006] proposed at the scientific competition *TRECVID* and explain its integration into our framework in section 3.

2 A framework to enhance video analysis

The *Java*-based research framework *AMOPA – Automated MOving Picture Annotator* is easily extensible and allows rapid prototyping of arbitrary process-driven workflow concepts, traditionally used in image processing. Figure 1 shows the framework and several components, which are available as open source projects.

The open source library *FFMPEG* is used to open, close, and access any supported kind of video streams. The project *Streambaby*[1] and its subcomponent *FFMPEG-Java* directly invoke the *C* functions from *Java* code via *Java Native Access* (JNA).

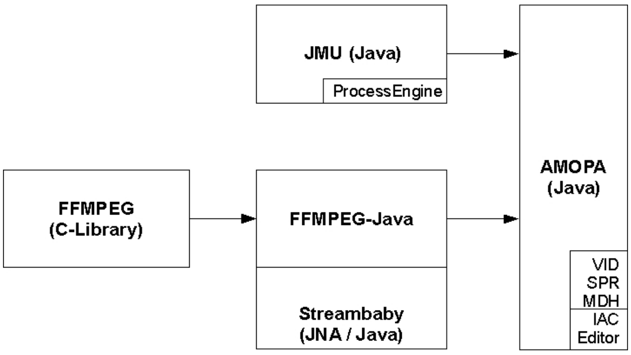


Figure 1: Architecture of the research framework *AMOPA*. (From: [Ritter, 2009])

The implementation of the process concept is based on an extended version of the toolkit *Java Media Utility*, provided in 2006 in an early state of development by the engineer Paolo Mosna [Mosna, 2007]. Linear workflows can be modeled by using its engine framework. Thereby connected processes allow to pass results in the form of defined objects along the *image processing chain* (IPC). Any chain contains one input, one output, and several intermediate processes, which can be dynamically instantiated and parameterized during runtime by the means of an *XML* file. All custom processes are started as separated full-functional threads with optional object sharing.

To facilitate and accelerate the creation of customized image processing chains, a visualization toolkit consisting of two components is currently under development. The first one is a graphical editor, oriented at the comfortable usability of *GraphEditPlus*[2]. The other one implements a global window manager, capable of handling any graphical output at any time from registered processes.

3 Applied shot detection

The international scientific *Text REtrieval Conference (TREC)* series “encourage[s] research in information retrieval from large text collections”[3]. The track on *Video Retrieval Evaluation* became an independent evaluation (*TRECVID*) in 2003. Finally, *Shot Boundary Detection* remained a major task until 2007. [Smeaton *et al.*, 2006]

[1] http://code.google.com/p/streambaby

[2] http://www.thedeemon.com/GraphEditPlus

[3] http://trec.nist.gov

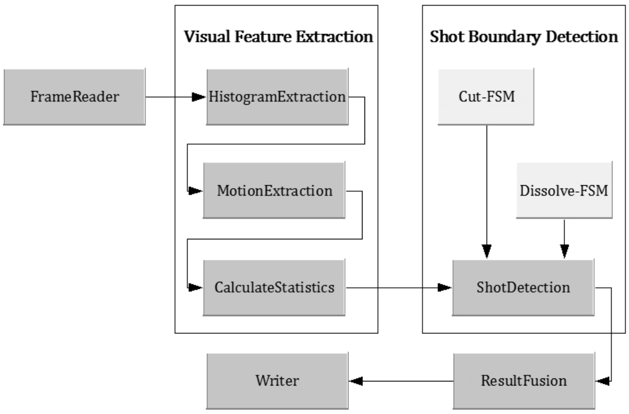


Figure 2: Adapted processing chain from the method of AT&T in *AMOPA*. All processes are marked in dark grey. The white finite state machine detectors Cut and Dissolve are simply aggregated by the connected controlling process used for automated shot detection.

3.1 Definitions

According to [Smeaton *et al.*, 1999] a *shot* is defined as a sequence of frames "resulting from continuous recording by a single camera", whereas a *scene* is frequently composed of multiple shots. In turn the video itself naturally consists of a collection of scenes.

Transitions between two adjacent shots can be divided into four major types. A *"hard" cut* spans two consecutive frames and occurs after a complete change of shots. Fade, dissolves and wipes are examples of gradual transitions containing more than two frames. [Zhang *et al.*, 1993]

3.2 The method of AT&T

[Liu *et al.*, 2006] suggested a promising solution to reliably detect different types of shot transitions. Since *hard cuts* and *dissolves* occur most frequent, we decided to implement the corresponding detectors within our framework for further analysis.

Figure 2 illustrates the adapted image processing chain of the related work from AT&T. The video frames are read by the input process *FrameReader* and are passed onto the feature extraction component (left box). The shot boundary detection is done by the connected detectors (right box), which are using the results from feature extraction. In the end the module *ResultFusion* avoids the overlapping of shot transition candidates.

The integrated block-based motion detection is one of the key features to detect shot boundaries for sure. At first the current image has to be segmented into non-overlapping blocks, preferably of size 48×48. Subsequently a motion vector for this template to the next frame is calculated using a customized search range of 32×32. The difference between the best match within the search range and the underlying template is called *matching error*. These statements are repeated for every block and the current overall matching error ME_A is computed.

The actual shot detection is performed by the shot detectors, which are implemented as finite state machines (*FSM*), starting at state 0. As an example, figure 3 shows the variables used in the cut detector along a sample video. The detector owns a state variable $AverageME$, that is updated by convex linear infinite impulse response in state 0. It changes into transition candidate state 2, if the $AverageME$ is a multiple of its predecessor and if the current ME_A is higher than within the last five frames. The verification state 3 is reached, if the current ME_A remains higher than before. If the dissimilarity between the shot candidates is high, a detected shot is marked in state 1. In any other cases state 0 is invoked.

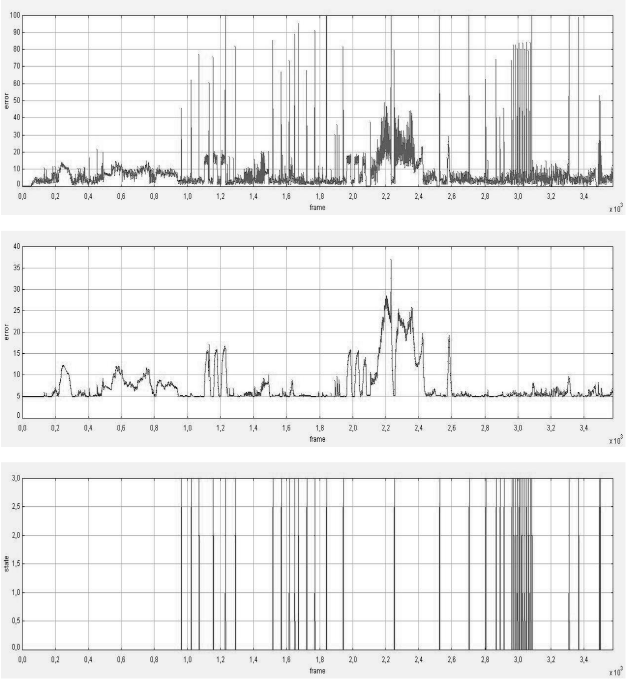


Figure 3: Visualization of the course of variables from the cut detector ($AverageME$—upper figure, ME_A—middle figure, detected cuts are represented by the peaks of the different states of the finite state machine within the lower figure) at the randomly chosen video BG_26797 from *TRECVID* 2008 data set with an overall length of 3,605 frames.

First runs indicate, that the hard cut detector seems to perform superior at data sets from different TRECVID years as well as from local TV stations.

4 Conclusions

Although our framework is still under development, we have shown in short, that it might be equipped with arbitrary algorithms. The chaining for the applied shot detection is not necessarily novel, but the possibility to select and visualize (parts of) algorithms at any stage in time and at comparatively low costs provides a convenient base for further development and examination of *state-of-the-art* algorithms. For more detail please refer to [Ritter, 2009].

A more sophisticated version of the presented algorithm from [Liu *et al.*, 2006] for shot detection was introduced by its authors in 2007, whereas a profound description can be found in [Liu *et al.*, 2007].

Acknowledgments

This work and the project *sachsMedia* was funded by *Unternehmen Region*, the initiative of innovations of the *BMBF Neue Länder*, Germany.

References

[Liu *et al.*, 2006] Zhu Liu, Eric Zavesky, David Gibbon, Behzad Shahraray, and Patrick Haffner. AT&T RESEARCH AT TRECVID 2006. Workshop Contribution, AT&T Labs-Research, 2006. http://www-nlpir.nist.gov/projects/tvpubs/tv6.papers/att.pdf, 13.05.2009.

[Liu *et al.*, 2007] Zhu Liu, Eric Zavesky, David Gibbon, Behzad Shahraray, and Patrick Haffner. AT&T RESEARCH AT TRECVID 2007. Workshop Contribution, AT&T Labs-Research, 200 Laurel Avenue South, Middletown, NJ 07748, 2007. http://www-nlpir.nist.gov/projects/tvpubs/tv7.papers/att.pdf, 13.05.2009.

[Mosna, 2007] Paolo Mosna. JMU: Java Media Utility, 2007. http://sourceforge.net/projects/jmu, 13.05.2009.

[Ritter, 2009] Marc Ritter. Visualisierung von Prozessketten zur Shot Detection. In *Workshop Audiovisuelle Medien: WAM 2009*, Chemnitzer Informatik-Berichte, pages 135–150. Chemnitz University of Technology, Saxony, Germany, 2009. http://archiv.tu-chemnitz.de/pub/2009/0095/index.html, 15.06.2009.

[sachsMedia, 2009] sachsMedia. InnoProfile Projekt sachsMedia — Cooperative Producing, Storage, Retrieval and Distribution of Audiovisual Media, 2009. http://www.tu-chemnitz.de/informatik/Medieninformatik/sachsmedia, http://www.unternehmen-region.de/de/1849.php, 14.05.2009.

[Smeaton *et al.*, 1999] Alan F. Smeaton, J. Gilvarry, G. Gormley, B. Tobin, S. Marlow, and N. Murphy. An evaluation of alternative techniques for automatic detection of shot boundaries. In *School of Electronic Engineering*, pages 8–9, 1999.

[Smeaton *et al.*, 2006] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press. http://www-nlpir.nist.gov/projects/trecvid/, 14.05.2009.

[Zhang *et al.*, 1993] Hongjiang Zhang, Atreyi Kankanhalli, and Stephen W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28, 1993.

Author Index

KDML 2009

Knowledge Discovery, Data Mining and Machine Learning

Editors

Dominik Benz, University of Kassel
Frederik Janssen, TU Darmstadt

Workshop on Knowledge Discovery, Data Mining, and Machine Learning (KDML) 2009

Dominik Benz
University of Kassel
Germany

Frederik Janssen
TU Darmstadt
Germany

The KDML Workshop

The workshop *Knowledge Discovery, Data Mining, and Machine Learning (KDML) 2009* is organized by the special interest group on Knowledge Discovery, Data Mining and Machine Learning (FG-KDML) of the German Society on Computer Science (GI).

The goal of the workshop is to provide a forum for database and machine learning oriented researchers with interests in knowledge discovery and data mining in order to discuss recent topics in these areas.

The workshop is part of the workshop week *Learning - Knowledge Discovery - Adaptivity (LWA) 2009* that also features several other workshops. This provides the opportunity to meet researchers from the related special interest groups on Adaptivity and Interaction, on Information Retrieval, and on Knowledge Management, and fosters (inter-workshop) scientific discussions and the important exchange of ideas.

KDML 2009

The program of this years KDML workshop consists of 14 regular and 6 short papers. It covers a broad range of technical and application papers, ranging from popular research topics such as text and community mining, multilabel classification and (bio-)medical applications.

The program committee received submissions from research and industry within the broad area of Knowledge Discovery and Machine Learning.

Special emphasis of this year's workshop was on submissions in the following areas:

- Mining and analysis of networks and graphs
- Text mining
- Web mining
- Distributed data mining and Ubiquitous knowledge discovery
- Unsupervised and Semi-supervised learning
- Visual analytics
- Bioinformatics applications
- Knowledge Discovery in inductive databases
- Data Stream Mining
- Temporal Knowledge Discovery
- Multi-criteria learning
- Rule Learning (Association rules/Inductive Rule Learning, Heuristics, Regression Rule Learning)

Program committee

The program committee had the following members (in alphabetical order):

- Martin Atzmüller, University of Würzburg
- Dominik Benz, University of Kassel
- Folke Eisterlehner, University of Kassel
- Zeno Gantner, University of Hildesheim
- Robert Jäschke, University of Kassel
- Frederik Janssen, TU Darmstadt
- Beate Krause, University of Kassel
- Eneldo Loza Mencía, TU Darmstadt
- Leandro Marinho, University of Hildesheim
- Sang-Hyeun Park, TU Darmstadt
- Jan-Nikolas Sulzmann, TU Darmstadt
- Lorenz Weizsäcker, TU Darmstadt

We would like to thank the authors for their submissions, and we also thank the members of the program committee for providing helpful constructive reviews.

September, 2009,

Dominik Benz and Frederik Janssen

Table of Contents

Fast and Effective Subgroup Mining for Continuous Target Variables

Martin Atzmueller, Florian Lemmerich, and Frank Puppe
Department of Computer Science, University of Würzburg, Germany
{atzmueller, lemmerich, puppe}@informatik.uni-wuerzburg.de

Abstract

Subgroup mining is a flexible data mining method that considers a given target variable and aims to discover interesting subgroups with respect to this property of interest. In this paper, we especially focus on the handling of continuous target variables: We propose novel formalizations of effective pruning strategies for reducing the search space, and we present the SD-Map* algorithm that enables fast subgroup mining for continuous target variables. Furthermore, we present effective visualization techniques that are seamlessly integrated in the semi-automatic subgroup mining process. The evaluation demonstrates the efficiency and effectiveness of the presented approaches.

1 Introduction

Subgroup mining is a broadly applicable data mining technique that can be customized for various application domains. Prominent examples include knowledge discovery in medical and technical domains, e.g., [Lavrac *et al.*, 2004; Atzmueller *et al.*, 2005]. In general, subgroup mining aims to identify *interesting* groups of individuals that deviate from the norm considering a certain property of interest [Wrobel, 1997] given by a target variable. For example, the risk of coronary heart disease (target variable) is significantly higher in the subgroup of smokers with a positive family history than in the general population.

Subgroup mining with continuous target variables has received increasing attention recently, e.g., [Jorge *et al.*, 2006; Aumann and Lindell, 2003], especially regarding industrial applications. For example, in the industrial domain often one important goal is the identification of subgroups described by a combination of certain factors that cause a significant increase/decrease in certain measurement parameters, e.g., the number of service requests for a certain technical component, or the fault/repair rate of a certain manufactured product. Then, the target subgroups can often not be analyzed sufficiently using the standard techniques, e.g., [Lavrac *et al.*, 2004; Wrobel, 1997], for binary/nominal target variables, since the discretization of the target variables causes a crucial loss of information. As a consequence, the interpretation of the results is often difficult. The development of the techniques presented in this work aims to overcome the mentioned limitations: Their development was mainly motivated by industrial applications in the service support and in the manufacturing domain that required fast responsiveness for interactive scenarios.

In this paper, we propose methods for fast and effective subgroup mining for continuous target variables and present an efficient subgroup discovery algorithm – as the core step of the subgroup mining process: Usually a set of (initial) hypotheses is discovered that is then refined by interactive or focused automatic methods incrementally. For a comprehensive analysis, the efficient automatic approaches are important for supporting a semi-automatic involvement of the domain experts in order to effectively contribute in a discovery session. Then, the discovered subgroups can be refined utilizing visualization techniques.

The contribution of the paper is thus twofold: We focus on pruning strategies for reducing the search space in order to obtain upper bounds for the possible quality of the discovered patterns. We also present the SD-Map* algorithm as a novel adaptation of the efficient SD-Map [Atzmueller and Puppe, 2006] algorithm incorporating the proposed pruning strategies. Similar to the SD-Map algorithm, SD-Map* guarantees to identify the k best patterns but it is significantly faster using the pruning techniques. Furthermore, we present effective visualization techniques and discuss how these are integrated in the data mining environment VIKAMINE (`http://www.vikamine.org`).

The rest of the paper is organized as follows: Section 2 provides the basics on subgroup mining. After that, Section 3 introduces the setting of continuous target variables and proposes novel implementations of tight optimistic estimate functions for the continuous case. Next, we propose the SD-Map* algorithm that enables fast subgroup mining for continuous target variables. After that, we introduce effective visualization methods, and discuss related work. Section 4 provides an evaluation of the approach demonstrating its efficiency and effectiveness. Finally, Section 5 concludes with a summary of the presented work and provides pointers for future work.

2 Preliminaries

In the following, we first introduce the necessary notions concerning the used knowledge representation, before we introduce subgroup discovery and its implementation using continuous target variables.

2.1 Subgroup Mining and Subgroup Discovery

The main application areas of subgroup mining are exploration and descriptive induction to obtain an overview of the relations between a (dependent) target variable and a set of explaining (independent) variables. Then, the goal is to uncover properties of the selected target population of individuals featuring the given target property of interest. Therefore, not necessarily complete relations but also partial relations, i.e., (small) subgroups with "interesting" characteristics can be sufficient. Specifically, these interesting subgroups should have the most unusual (distributional) characteristics with respect to the concept of interest given by the target variable [Wrobel, 1997]. Subgroup discovery is the core data mining step of the subgroup mining process, e.g., [Atzmueller *et al.*, 2005], that is usually implemented using automatic (algorithmic) and interactive techniques, e.g., visualization methods.

A subgroup discovery task mainly relies on the following four main properties: the target variable, the subgroup description language, the quality function, and the discovery strategy. Since the search space is exponential concerning all the possible selectors of a subgroup description efficient discovery methods are necessary.

For some basic notation, let Ω_A denote the set of all attributes. For each attribute $a \in \Omega_A$ a range $dom(a)$ of values is defined. Let CB be the case base (data set) containing all available cases (instances). A case $c \in CB$ is given by the n-tuple $c = ((a_1 = v_1), \ldots, (a_n = v_n))$ of $n = |\Omega_A|$ attribute values, $v_i \in dom(a_i)$ for each a_i.

The subgroup description language specifies the individuals belonging to the subgroup. For a commonly applied single-relational propositional language a subgroup description can be defined as follows:

Definition 1 (Subgroup Description). *A subgroup description $sd(s) = \{e_1, \ldots, e_n\}$ of the subgroup s is defined by the conjunction of a set of selection expressions (selectors). The individual selectors $e_i = (a_i, V_i)$ are selections on domains of attributes, $a_i \in \Omega_A, V_i \subseteq dom(a_i)$. We define Ω_E as the set of all selection expressions and Ω_{sd} as the set of all possible subgroup descriptions.*

A subgroup s described by $sd(s)$ is given by all cases $c \in CB$ covered by the subgroup description $sd(s)$. A subgroup s' is called a *refinement* of s, if $sd(s) \subset sd(s')$.

2.2 Subgroup Quality Functions

A quality function measures the interestingness of the subgroup and is used to rank these. Typical quality criteria include the difference in the distribution of the target variable concerning the subgroup and the general population, and the subgroup size.

Definition 2 (Quality Function). *Given a particular target variable $t \in \Omega_E$, a quality function $q : \Omega_{sd} \times \Omega_E \to R$ is used in order to evaluate a subgroup description $sd \in \Omega_{sd}$, and to rank the discovered subgroups during search.*

One major difference between the different types of target variables is given by disjoint sets of applicable quality functions due to the different parameters that can be applied for estimating the subgroup quality: For example, target *shares* are only applicable for binary or categorical attributes, while continuous target variables require averages/aggregations of values, e.g., the *mean*. In the following, we consider the functions *Continuous Piatetsky-Shapiro* q_{CPS} (adapted from [Grosskreutz *et al.*, 2008]), *Continuous LIFT* q_{CLIFT}, and *Continuous Weighted Relative Accuracy* q_{CWRACC} (see [Lavrac *et al.*, 2004]):

$$q^c_{Wracc} = \frac{n}{N} \cdot (m - m_0)\,,\; q^c_{PS} = n \cdot (m - m_0)\,,\; q^c_{Lift} = \frac{m}{m_0}\,.$$

where n and N denote the size of the subgroup and the size of the total population as defined above, respectively, and m specifies the mean of the target variable within the subgroup; m_0 specifies the mean of the target variable in the total population. $n \geq \mathcal{T}_n$ specifies a minimal size constraint for the subgroup.

The *CN2-SD* algorithm [Lavrac *et al.*, 2004], is a prominent example of an heuristic subgroup discovery algorithm that applies a beam-search strategy. The adaption of such an algorithm is rather simple, as in each step the quality values of the subgroup hypotheses contained in the beam are directly updated from the case base. Instead of determining the target share(s) of (binary) target variables, simply the mean values of the cases contained in the subgroup m and (once) for the total population m_0 need to be obtained. It is easy to see that the continuous case subsumes the binary one as a special case: Computing the averages includes computing the target shares – by considering the values 1 and 0 for a *true/false* target concept, respectively. The ordinal case is captured by mapping the ordinal values to continuous values and normalizing these if necessary.

3 Adapting Subgroup Mining for Continuous Target Variables

In the following, we show how to efficiently adapt exhaustive subgroup mining for continuous target variables: In contrast to heuristic approaches, exhaustive subgroup discovery algorithms guarantee to identify the k best subgroups. We discuss tight optimistic estimate quality functions [Grosskreutz *et al.*, 2008], and we introduce novel formalizations for the case of continuous target variables. Additionally, we present the SD-Map* algorithm that enables efficient subgroup mining using tight optimistic estimates for the continuous case. Finally, we discuss novel visualization techniques, show how the presented approaches are integrated into the subgroup mining process, and discuss related work.

3.1 Tight Optimistic Estimates

The basic idea of optimistic estimates [Grosskreutz *et al.*, 2008] is to safely prune parts of the search space. This relies on the intuition that if the k best hypotheses so far have already been obtained, and the optimistic estimate of the current subgroup is below the quality of the worst subgroup contained in the k best, then the current branch of the search tree can be safely pruned. More formally, an optimistic estimate oe of a quality function qf is a function such that $s' \subseteq s \Rightarrow oe(s) > qf(s')$, i.e., that no refinement of subgroup s can exceed the quality $oe(s)$. An optimistic estimate is considered *tight* if for any database and any subgroup s, there exists a subset $s' \subseteq s$, such that $oe(s) = qf(s')$. While this definition requires the existence of a subset of s, there is not necessarily a subgroup description, that describes s', cf., [Grosskreutz *et al.*, 2008].

For binary targets the determination of such a best subset is relatively simple using any quality function that follows the monotony requirements for rule evaluation functions postulated in [Klösgen, 1996]. The best subset is always given by the set of all cases, for which the target concept is true. We introduce the following notation: $n(s) = |\{c \in s\}|$ specifies the size of subgroup s, $tp(s) = |\{c \in s | t(c) = true\}|$ the number of positive examples in s; $t(c)$ denotes the value of the target variable in case c and $p(s) = \frac{tp(s)}{n(s)}$ is the target share of the subgroup.

Theorem 1. *For each subgroup s with $p > p_0$ and for each boolean quality function q for which the axioms postulated in [Klösgen, 1996] apply: $s' \subseteq s \Rightarrow q(s') \leq q(s^*)$, where $s^* = \{c \in s | t(c) = true\}$*

Proof. We first show, that $q(s) \leq q(s^*)$. This means, that the quality of any subgroup with a positive quality is always lower or equal to the quality of the subset of examples, that only contains the positive examples of s. We apply the third axiom of [Klösgen, 1996]: "$q(s)$ monotonically decreases in n, when $p = c/n$ with a fixed constant c.": As fixed constant c we consider the number of positive examples tp, as $p = c/n \Leftrightarrow c = p \cdot n$ and $n(s) \cdot p(s) = tp(s) = tp(s^*) = n(s^*) \cdot p(s^*)$. So, the quality function monotonically decreases in n. As $n(s) > n(s^*)$ we conclude: $q(s) \leq q(s^*)$.

For arbitrary $s' \subseteq s$ we now need to consider two cases: If $s^* \subseteq s'$, then $q(s') \leq q(s^*)$ as shown above. If $s^* \not\subseteq s'$, then there exists a subset $s'' = \{c \in s | t(c) = true\}$, that contains only the positive examples of s'. The above proof then implies, that the quality of this subset is at least as high as the quality of its superset s: $q(s') \leq q(s'')$. On the other hand $s'' \subseteq s^*$ is true, as s'' only consists of positive examples of s. We now apply the fourth axiom of [Klösgen, 1996]: "q(s) monotonically increases in n when $p > p_0$ is fixed.": As $p(s'') = p(s^*) = 1$ it follows that $q(s'') \leq q(s^*)$. Thus, $q(s') \leq q(s'') \leq q(s^*)$, proving the theorem. □

Thus, in the binary case the subset of a subgroup with the highest quality can be easily found, since it consists of the same cases for any well-formed quality function. In contrast, the continuous case is more challenging, since the best subset of a subgroup is dependent on the used quality function. Consider the following example with an average target value of $m_0 = 50$ and the subgroup s containing cases with values 20, 80 and 90. Then, for the quality function q_{CLIFT} with $\mathcal{T}_n = 1$ the subset with the best quality contains only the case with value 90. On the other hand for the Pietatsky-Shapiro quality function, it contains two cases with the values 80 and 90, respectively. However, the following theorems provide easy to compute tight optimistic estimates for several important continuous quality functions.

Theorem 2. *For the Piatetsky-Shapiro quality function $q_{CPS}(s) = n \cdot (m - m_0)$ the tight optimistic estimate for any subgroup s is given by*

$$oe(s) = \sum_{c \in s, t(c) > 0} (t(c) - m_0),$$

.

Proof. We reformulate the Piatetsky-Shapiro quality function:

$$\begin{aligned} q_{CPS}(s) &= n \cdot (m - m_0) \\ &= n \cdot (\frac{\sum_{c \in s} t(c)}{n} - \frac{n \cdot m_0}{n}) \\ &= \sum_{c \in s} t(c) - n \cdot m_0 \\ &= \sum_{c \in s} (t(c) - m_0) \end{aligned}$$

For all subsets $s' \subseteq s$, this sum reaches its maximum for the subgroup s^*, that contains all cases with larger target values than the average of the population, since it contains only positive summands, but no negatives. The quality of s^* is given by $q_{CPS}(s^*) = oe(s)$ using the above formula. As no other subset of s can exceed this quality the $oe(s)$ is an optimistic estimate. Since for any subgroup the estimate is obtained by one of its subsets, the estimate is tight. □

Please note, that the tight optimistic estimate for the binary case provided in [Grosskreutz *et al.*, 2008], i.e., $np(1 - p_0)$, can be seen as special case of this formula, considering $t(c) = 1$ for *true* target concepts and $t(c) = 0$ for *false* target concepts:

$$\begin{aligned} oe(s) &= \sum_{c \in s, t(c) > 0} (t(c) - m_0) \\ &= \sum_{c \in s, t(c) = 1} (1 - p_0) \\ &= np(1 - p_0). \end{aligned}$$

Theorem 3. *Considering the quality function Weighted Relative Accuracy $q_{CWRACC}(s) = \frac{n}{N} \cdot (m - m_0)$ the tight optimistic estimate for any subgroup s is given by*

$$oe(s) = \frac{1}{N} \sum_{c \in s, t(c) > 0} (t(c) - m_0),$$

where t(c) is the value of the target variable for the case c.

Proof. q_{CWRACC} differs by the factor $\frac{1}{N}$ from the Pietatsky-Shapiro function. The population size can be considered as a constant, so the proof proceeds analogously. □

Theorem 4. *For the quality function Lift with a minimum subgroup size $\mathcal{T}_n$ the optimistic estimate is given by $oe(s) = \sum_{i=1}^{\mathcal{T}_n} (v_i - m_0)$, where v_i is the i-th largest value in the subgroup with respect to the target variable.*

Proof. Since the size of the subgroup is not relevant for these quality functions, the best possible subset is always the subset with the highest average of the target attribute with size k. The quality of this subset is given by the above formula. □

3.2 Fast Subgroup Mining with SD-Map*

SD-Map [Atzmueller and Puppe, 2006] is based on the efficient FP-growth [Han *et al.*, 2000] algorithm for mining frequent patterns. FP-Growth is based on the idea of utilizing a frequent pattern tree (FP-tree) which is implemented as an extended prefix-tree-structure that stores (extended) count information about the subgroup patterns and the relevant parameters for estimating their quality. The FP-tree contains the frequent nodes in a header table, and links to all occurrences of the frequent selectors in the FP-tree structure. This data structure itself can be regarded as a compressed data representation for the set of cases/instances. According to the prefix-tree principle, the tree stores aggregated counts for each shared path corresponding to the attribute–value pairs of a set of instances.

SD-Map utilizes an pattern-growth method similar to FP-growth with adaptations to the subgroup discovery setting: SD-Map applies a divide and conquer method, first mining subgroup patterns containing one selector and then recursively mining patterns of size 1 conditioned on the occurrence of a (prefix) 1-selector. For the recursive step, a conditional FP-tree is constructed, given the conditional pattern base of a set of frequent selectors (nodes): The conditional FP-tree stores all the instances that contain the conditioning selector(s), i.e., the conditional pattern base consists of all the prefix paths of the current conditioning node. Due to the limited space we refer to Han et al. [Han *et al.*, 2000] for more details on the FP-Growth method.

SD-Map utilizes the FP-tree structure built in two scans of the database in order to compute the qualities of subgroup patterns efficiently. For the binary case, an FP-tree node stores the subgroup size and the true positive count of the respective subgroup description. In the case of a continuous target variable, we need to consider the sum of the values of the target variable; these can also simply be stored in the nodes of the tree, enabling us to compute the respective quality functions value accordingly.

It is easy to see, that in this case all the necessary information is locally available in the FP-tree structure, i.e., within the currently considered node and its prefix paths contained in the tree. This feature enables the efficient computation of the respective quality parameters; for continuous target variables and optimistic estimate functions we consider further adaptations that are described in more detail below. The adaptions for continuous target variables even includes the case of a binary variable as a special case, where the value of the target variable is 1, if the target concept is *true* and 0, otherwise. More details on SD-Map are available in [Atzmueller and Puppe, 2006].

SD-Map* extends SD-Map by including (optional) pruning strategies and utilizes quality functions with tight optimistic estimates for this purpose: For embedding (tight) optimistic estimate pruning into the SD-Map algorithm, we basically only need to consider strategies for pruning the search tree and for reordering/sorting our hypotheses optimally according to the current (tight) optimistic estimates:

1. **FP-Header Pruning**: When building the frequent header of a (conditional) frequent pattern tree, we can omit all the nodes with an optimistic estimate below the minimum quality of the k best subgroup hypotheses obtained so far.
2. **FP-Tree Pruning**: When building a conditional FP-tree, we omit a (conditioned) branch, if the optimistic estimate for the conditioning selector is below the threshold given by the k best subgroup qualities.
3. **Optimistic Reordering**: During the iteration on the currently active selector queue contained in the header of a conditional) FP-tree, we can dynamically reorder the selectors that have not been evaluated so far by their optimistic estimate value. This way, we evaluate the *more promising* selectors first. This heuristic can help to obtain higher values for the pruning threshold early in the process, a way to prune more often earlier. Additionally, this step implements a modified depth-first search guided by the current optimistic estimates.

To efficiently compute the (tight) optimistic estimates we store additional information in the nodes of the FP-Tree, depending on the used quality function. For example, for the Piatetsky-Shapiro quality function we add the value $max(0, t(c) - p_0)$ for each case c to a field in the respective node during the construction of the FP-Tree. This field can also be propagated recursively – analogously to the sum of target values when building the conditional trees. It directly reflects the optimistic estimate of each node and can be immediately evaluated whenever it is needed for pruning. For the Lift quality function the adaptions are slightly more elaborate: During the creation of the frequent header nodes and the initial tree buildup, we save a list of the best $\mathcal{T}_n$ values in each node. When building the conditional trees, these value lists are merged, i.e., the best $\mathcal{T}_n$ are selected from the union of lists contributing to a node in the new conditional tree.

3.3 Visualization Techniques

In order to select a representative set of subgroups, visualization methods are usually a key technique for identifying the most interesting subgroups. After a set of subgroups has been discovered using automatic methods, they usually need to be assessed by the user in order to obtain the final set of subgroups that are really interesting and relevant.

The visualization techniques can be applied both for the interactive subgroup mining step directly – by guiding the user into the right or *interesting* direction for further exploration and subgroup refinement. Additionally, the visualizations can be used for a comprehensive assessment and comparison of the discovered subgroups. While the *Bar Visualization* and the *Zoomtable Visualization* are mostly used for the interactive step directly, the *M/N Diagram* and the *Distribution Histogramm Visualization* are applied for the post-processing, assessment and refinement of a set of subgroups. In the following sections, we discuss the different visualization methods in detail. The visualization techniques are implemented in the VIKAMINE (Visual, Interactive and Knowledge-Intensive Analysis and Mining Environment) system [Atzmueller, 2007]. VIKAMINE is a versatile tool for intelligent data mining, that already provides a rich set of discovery, analysis, and visualization options. Our experiences in various application projects have shown, that the visualizations are essential in order to provide user support during the result assessment and refinement phase of the data mining process.

Zoombar Visualization

The zoombar visualization (see Figure 1) gives a fast overview on the most important data of a subgroup in relation to the total population, and shows the current mean values and the respective subgroup/population sizes. It is used during the interactive subgroup mining step when assessing the current subgroup hypothesis. The visualization consists of two bars: The upper bar represents the total population, the lower bar represents the current subgroup. The total width of each bar shows their respective size. Each bar is split in green and a red part in proportion to the average value of the target variable. The larger the green bar is, the higher is the average value of the target attribute. To obtain the ratio of areas parts the domain of the target value is normalized to the interval $[0; 1]$ and the average target value is mapped accordingly. Thus, the green bar shows the sum of all normalized target values in the subgroup or total population.

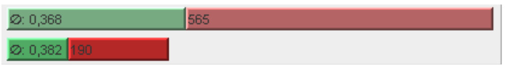


Figure 1: Exemplary zoombar visualization

The bar visualization has a rather nice property for assessing the respective target distributions: If the target value starts at 0, which is the case in many application scenarios, then the green bar indicates the sum of all target values. Therefore, the green area of the upper bar is always at least as large as the green area of the lower bar. If both green areas have the same size all target values above the domain minimum are included in the selected subgroup. For the (subsumed) special case of a boolean target concept the size of the green bar reflects the number of positive cases in the subgroup.

For example, we could consider the scrap rate of some industrial production process as target variable with the domain [0; 1]. Then, the scrap rate for the subgroup/total population is shown by the proportion of the green part in the respective bar. The share of faulty parts covered by the cases in the subgroup can be easily observed by comparing the size of the green parts (given a constant production in all cases). If the size of the lower green bar reaches the size of the upper green bar, then all faulty parts are produced in the subgroup including these cases.

Zoomtable Visualization

The zoomtable, see Figure 2, shows the distribution of the data restricted to a selected subgroup: each row of the zoomtable shows the value distribution of a specific attribute limited to the cases covered by the current subgroup; the width of each cell relates to the frequency of the respective attribute value. Additionally, a cell of the zoomtable shows detailed information for further exploration and subgroup refinement concerning the target variable. The user can then perform interactive subgroup discovery guided by the information contained in the respective cells.

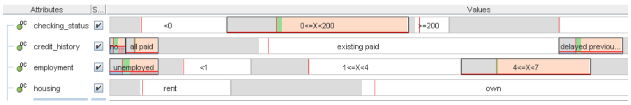


Figure 2: Example of the zoomtable visualization for continuous target variables (credit-g dataset)

For a detailed view, Figure 3 shows the abstract structure of a row of the zoomtable including the type of the attribute, its current ranking, the attribute name, and its value distribution annotated with several visual markers. Two of the most important parameters with respect to a continuous target variable are the *mean within the subgroup* (m) and the *subgroup size* (n) of the respective subgroup, and their deviations from the general population given by all cases of the database. There is usually a trade-off between these parameters that is formalized by the quality function.

The *subgroup size* with respect to a future subgroup is given by the width of a specific selector cell. The current target mean is visualized in the individual cells by visual markers.

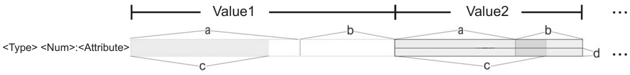


Figure 3: The zoomtable visualization for continuous target variables – detailed view

With respect to the given current subgroup s, (a) indicates the normalized mean of the general population, while (b) denotes the remainder; (c) indicates the normalized mean of the target variable for the subgroup s', i.e., the subgroup that is constructed by including the particular attribute value. If (c) is larger than (a), then the *target mean* increases adding this selector. Then, (d) shows the relative gain in the target mean m, comparing the subgroups s and s', i.e., $d \sim \frac{c-a}{b}$, $c \geq a$, for an easier assessment of small cells. If the height of (d) is zero, then the target mean does not increase. If it fills the entire bar, then the target share reaches 100%.

By interpreting these visual markers which are shown using different colors the user can immediately identify promising improvements of the currently active subgroup. Furthermore – if enabled – the zoomtable ranks the rows of the table with the most significant improvement, shown by the number in the column left to the value cells.

By selecting selectors that are marked as interesting in the zoomtable, the user can manipulate the current subgroup by one click selecting cells in the zoomtable; the zoomtable is animated and updated immediately with respect to the respective value distributions. Then, interactive exploration can be performed very easily and effectively.

M/N Diagram

For a comparative and intuitive presentation of a set of subgroups, we propose the *m/n-diagram*, see Figure 4 for an example. In this visualization each subgroup is represented by exactly one point in the diagramm. The size of a subgroup determines its position on the x-axis, while the average target value corresponds to its position on the y-axis. Additionally, the average target value in the total population is marked by a red line, in parallel to the x-axis.

Considering this visualization, the quality of the subgroup with respect to the continuous Piatetsky-Shapiro quality function is thus visualized by the area size of the rectangle with the subgroups position as the upper right corner and the intersection of the red average line and the y-axis as lower left corner. Additionally, the lift quality value of the subgroup is given by its position on the y-axis.

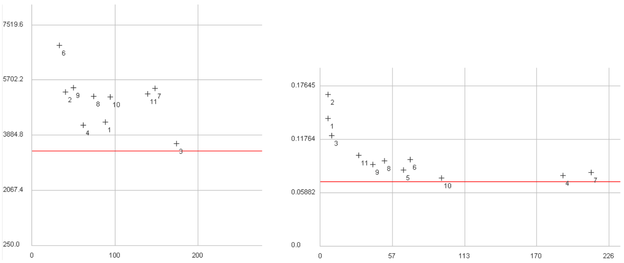


Figure 4: Basic M/N Diagram: The subgroups labeled 2, 1, and 3 in the right diagram are relatively small, but differ significantly from the total population with respect to the target variable. In contrast, the subgroups labeled 4 and 7 only show a small increase with respect to the target variable but are quite large. The subgroup description can be obtained via the description panel which is not shown in the screenshot. A single click on a subgroup enables further analysis and introspection options.

Since often many important subgroups have a relatively small size compared to the total population we propose to utilize a non linear scaling for the x-axis (e.g., $\sqrt{n}$ instead of n) as an alternative. This enlarges the important areas on the left of the diagramm. If these become to densely populated, i.e., displaying too many subgroup patterns, then the user can easily choose between the suitable level of detail.

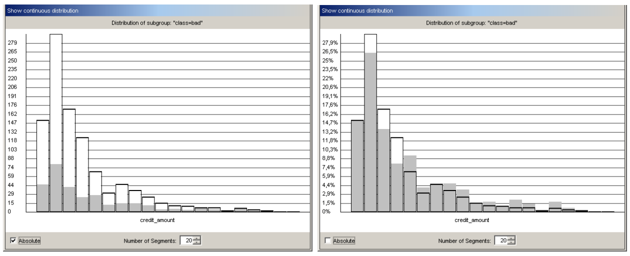
Distribution Histogramm Visualization

A visualization especially important for interactive subgroup mining with a continuous property of interest is the distribution histogramm (see Figure 5). This visualization allows for an easy comparison of the distribution of the target variable in the subgroup and the total population.

The visualization contains a bar chart: The gray bars describe the distribution of the target variable in the subgroup, the white bars (outlined within black borders) display the distribution of the target variable with respect to the total population. The x-axis shows the target values discretized in small segments, the y-axis (height of the bars) the absolute number of cases with a target value within the respective interval. Thus, the gray (subgroup) bar is always smaller than the white bar. If both bars are equally large, then all cases with a target value in the segment described by the respective bar are part of the subgroup.

A variation of this visualization that we found especially useful for inspecting small subgroups uses the proportion of cases in the interval in relation to all cases in the subgroup or population instead of the absolute counts. In this variant, the analyst can even more easily determine the increase/decrease in parts of the distribution of the subgroup. However, no information about the coverage of cases by the subgroup in the segments is given, in contrast to the basic visualization.

Optionally, the visualization provides further visual markers: If enabled, bars with an increasing target mean are marked in green color, while bars with a decreasing target mean are shown in red color, comparing the subgroup and the population, respectively.



☑ Absolute Number of Segments: 20

□ Absolute Number of Segments: 20

Figure 5: Distribution Histogramm: This example shows the target variable credit_amount with respect to the subgroup ’class = bad’. The left figure shows the absolute distribution, the right figure shows the relative counts with respect to the size of the subgroup. We can easily observe in the diagram showing the relative counts, that large credit amounts are more often requested by customers classified as ’bad’.

3.4 Related Work and Discussion

In this paper, we proposed novel formalizations of tight optimistic estimates for numeric quality functions. Additionally, we presented the SD-Map* algorithm that enables efficient subgroup mining for continuous target variables. By utilizing these novel quality functions, SD-Map* shows a significant decrease in the number of examined states of the search space, and therefore also a significant reduction concerning the runtime and space requirements of the algorithm, as shown in the evaluation in Section 4.

Additionally, we introduced novel visualization techniques for subgroup patterns with continuous target variables that allow an easy assessment and comparison of a set of the discovered subgroups. The zoombar and the zoomtable visualizations are adaptations from visualizations for binary/nominal target variables [Atzmueller, 2007], and were originally inspired by the InfoZoom [Spenke, 2001] system that also utilizes bar visualizations showing the distributions of variables. In contrast, the presented visualizations are significantly enhanced using the described visual annotations that directly indicate how *promising* or interesting the respective refinements really are. The histogram visualization is an adaptation of a basic data analysis technique with special focus on the comparison of the relative/absolute target values of the subgroup/population groups.

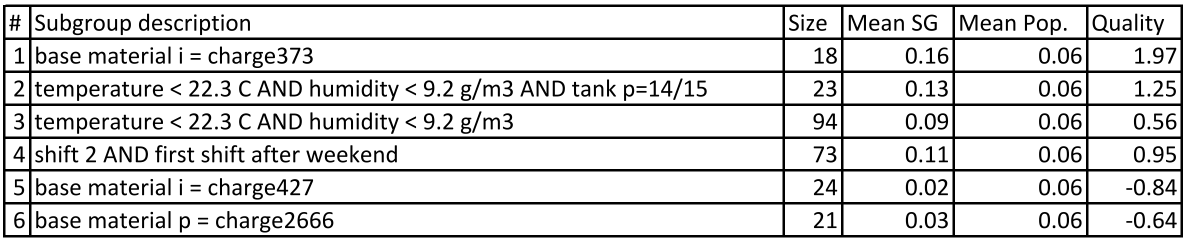
Handling continuous target variables in the context of subgroup mining has been first discussed by Kloesgen [Klösgen, 1996] in the EXPLORA system. Kloesgen applied both heuristic and exhaustive subgroup discovery strategies without pruning. An improvement was proposed by Wrobel [Wrobel, 1997], presenting optimistic estimate functions for binary target variables. Recently, Grosskreutz et al. [Grosskreutz *et al.*, 2008] introduced tight optimistic estimate quality functions as a further improvement on optimistic estimate quality functions for binary and nominal target variables. Additionally, Grosskreutz et al. introduced the DpSubgroup algorithm that also incorporates tight optimistic estimate pruning. While their algorithm is somehow similar to the SD-Map algorithm, since also a frequent pattern tree is used for efficiently obtaining the subgroup counts, the DpSubgroup algorithm focuses on binary and categorical target concepts only, and lacks the efficient propagation method of SD-Map* when computing the tight optimistic estimates for continuous target variables in the FP-tree directly. Furthermore, SD-Map* is applicable for binary, categorical, and continuous target variables. DpSubgroup also uses an explicit depth-first search step for evaluating the subgroup hypotheses while this step is implicitly included in the divide-and-conquer frequent pattern growth method of SD-Map* directly (that is, by the reordering/sorting optimization).

Jorge et al. [Jorge *et al.*, 2006] introduced an approach for subgroup mining with continuous target variables applying special visualization techniques for the interactive discovery step. In contrast to the presented approach, the methods focus on interactive techniques for identifying distribution rules, as a special case of subgroup patterns: These apply quality functions based on goodness-of-fit tests for measuring the deviations of the target variable averages in the subgroup vs. the total population. Therefore, the presented approach is more general since it includes arbitrary quality functions for continuous target variables.

Furthermore, Jorge et al. apply an adapted algorithm for discovering frequent sets, so there are no pruning options for enabling a more efficient discovery process. Considering the visualization options, we presented a set of visualizations that provide both options for aggregated and detail analysis of the discovered subgroups for post-processing *and* for the discovery/refinement: While the zoomtable and the zoombar visualization support the interactive discovery step, m/n-diagramm and the histogram-visualization provide powerful means for the post-processing of a set of subgroups. In constrast, Jorge et al. only focus on the post-processing options considering the distributions of the target variable.

4 Evaluation

We provide an evaluation utilizing real-world data from the industrial domain using the novel SD-Map* algorithm with pruning and the ’standard’ SD-Map algorithm as a reference; for both, we apply the continuous Piatetsky-Shapiro quality function. For the evaluation, we consider two data sets: The applied industrial data set contains about 20 attributes and a sample of about 1000 cases from the orig-



#	Subgroup description	Size	Mean SG	Mean Pop.	Quality
1	base material i = charge373	18	0.16	0.06	1.97
2	temperature < 22.3 C AND humidity < 9.2 g/m3 AND tank p=14/15	23	0.13	0.06	1.25
3	temperature < 22.3 C AND humidity < 9.2 g/m3	94	0.09	0.06	0.56
4	shift 2 AND first shift after weekend	73	0.11	0.06	0.95
5	base material i = charge427	24	0.02	0.06	-0.84
6	base material p = charge2666	21	0.03	0.06	-0.64

Figure 6: Exemplary results using the industrial data set for the target variable *scrap rate*. The table shows the factors describing the different subgroups (subgroup description), the means of the target variable in the subgroup and the total population, and the quality of the subgroups as measured by the relative gain quality function.

inal industrial data [1]. The data set contains parameters of a manufacturing process, e.g., certain types of chemical base materials, information about the active shift, and measurements like temperature and the pressure in certain machines. The output of the process is categorized as *ok*, *scrap*, *needs repair*. The obvious goal is to decrease the scrap and repair rates in order to increase the effectiveness of the process and to decrease costs. Then, either 'good' parameters (causing low scrap/repair rates), or 'bad' parameters (causing high rates) can be identified.

Figure 6 shows some exemplary results of the discovery process that was implemented using the presented approach. The table shows both subgroups (as combinations of factors) that cause an increased scrap rate, but also factors that help to lower the scrap rate. In this sense, the discovered factors can be either used for training or for measures that change the production conditions. Rows 1-4 show the 'negative' factors that are correlated with an increased scrap rate, therefore the subgroup quality is positive. Rows 5-6 show the 'positive' factors, that is, the factors that are associated with a lower scrap rate. In summary, the semi-automatic process enabled rather short incremental cycles due to the utilization of the efficient discovery method SD-Map*. Combined with the visualization methods, this approach supported the data analysts and domain experts significantly.

The second data set that we applied for the evaluation mainly for measuring the scalability of the approach is the credit-g data set available from the UCI [Newman *et al.*,] repository; it contains about 1000 cases; the data set describes customer ratings categorized whether these have a *good* or *bad* credit rating.

Figure 7 shows the number of hypotheses that were considered during the discovery process. As discussed above, the complexity grows exponentially with the number of attributes and attribute values. It is easy to see, that the optimistic estimate pruning approach shows a significant decrease in the number of hypotheses considered, and thus also in the runtime of the algorithm. The 'full' optimistic estimate pruning approach using dynamic reordering strategy (optimistic estimate pruning and dynamic sorting of the FP-Tree-header nodes) also shows an additional improvement by a factor of two compared to the approach using only optimistic estimate pruning.

Figure 8 shows the results of applying the approach on the credit-g data set from the UCI [Newman *et al.*,] repository. We considered the target concept *credit amount*, and supplied the nominal attributes of the data set subsequently. The results confirm the observation for the industrial data, and the decrease of considered hypotheses/running time is even more significant. Similar to the industrial data set, the 'full' pruning/sort strategy of SD-Map* shows slight 'variations' with respect to the number of considered hypotheses (cf., lines for 9 and 12 attributes of credit-g): This can be explained by the fact that the ordering strategy can yield better subgroup qualities earlier in the process.

These results clearly indicate the benefit and broad applicability of the approach: The pruning strategies enable fast (automatic) subgroup discovery for continuous target variables, that can then be a starting point for a detailed analysis. Further analysis options are then given by the visualization techniques, and due to the efficiency of the automatic methods, a seamless integration with short 'roundtrip' times during analysis sessions can be accomplished.

5 Conclusions

In this paper, we have presented an approach for fast and effective subgroup mining focusing on continuous target variables. The presented techniques feature novel formalizations of tight optimistic estimate quality functions and the SD-Map* algorithm for enabling efficient subgroup mining for continuous target concepts. Additionally, we have proposed several visualization techniques for effective subgroup discovery. These enable an intuitive and seamless application of the presented techniques, and the integrated approach has been implemented in the data mining environment VIKAMINE, e.g., [Atzmueller, 2007], available at `http://www.vikamine.org`.

The evaluation of the algorithmic approach showed significant improvements concerning the efficiency of the subgroup discovery method since large areas of the search space could be safely pruned. This enables a transparent application of the algorithm even in rather interactive contexts. For these, the powerful visualization techniques also proved essential for supporting the application of a semi-automatic discovery process.

For future work, we aim to assess a combination of tight (continuous) optimistic estimates with sampling techniques and methods for distributed subgroup discovery in order to optimize the efficiency of the subgroup discovery method even more. Additionally, we plan to extend the presented subgroup mining approach to time-oriented sequence data in medical and technical domains.

Acknowledgements This work has been partially supported by the German Research Council (DFG) under grant Pu 129/8-2.

[1] Unfortunately we cannot make this dataset publicly available due to non-disclosure reasons.

#Attributes	w/o OE-Pruning	w/OE-Pruning	w/OE-Sort-Pruning
2	23	23	23
3	83	61	50
4	117	57	41
5	157	67	48
6	390	101	55
7	464	109	50
8	544	112	43
9	1546	131	48
10	2763	181	69
11	4333	170	84
12	4486	173	87
13	7567	219	110
14	9123	275	155
15	14057	325	186
16	20112	432	240
17	29708	548	335
18	43517	583	390
19	58139	612	381
20	65307	800	469

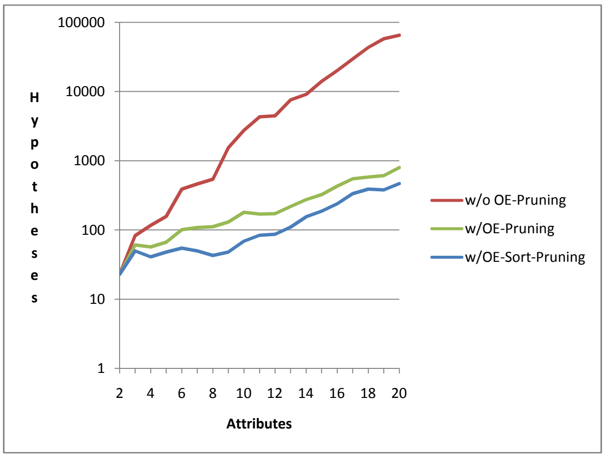


Figure 7: Evaluation: Industrial Data. The x-axis of the graph shows the number of attributes provided to the discovery method, the y-axis shows the resulting number of considered hypotheses. The columns of the table indicate the number of hypotheses without pruning (*w/o OE-Pruning*), with optimistic-estimate pruning and no sorting (*w/ OE-Pruning*), and with the full optimistic-estimate/sorting strategy (*w/OE-Sort-Pruning*).

#Attributes	w/o OE-Pruning	w/OE-Pruning	w/OE-Sort-Pruning
2	27	27	27
3	215	119	68
4	652	130	102
5	2019	181	158
6	4756	238	170
7	7342	240	177
8	16779	260	168
9	28770	347	212
10	50015	411	307
11	93626	448	343
12	168818	500	212
13	224087	558	229

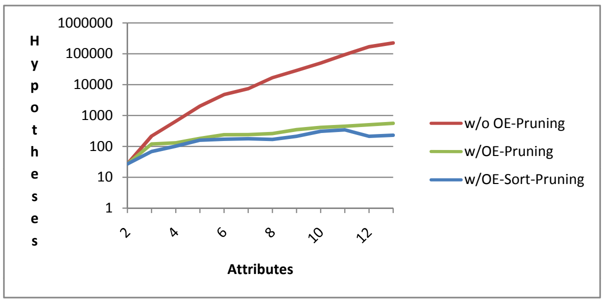


Figure 8: Evaluation: Credit-G Data Set. The x-axis of the graph shows the number of attributes provided to the discovery method, the y-axis shows the resulting number of considered hypotheses (see Figure 7 for the used parameters).

References

[Atzmueller and Puppe, 2006] Martin Atzmueller and Frank Puppe. SD-Map – A Fast Algorithm for Exhaustive Subgroup Discovery. In *Proc. 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2006)*, number 4213 in LNAI, pages 6–17, Berlin, 2006. Springer.

[Atzmueller *et al.*, 2005] Martin Atzmueller, Frank Puppe, and Hans-Peter Buscher. Exploiting Background Knowledge for Knowledge-Intensive Subgroup Discovery. In *Proc. 19th Intl. Joint Conf. on Artificial Intelligence (IJCAI-05)*, pages 647–652, 2005.

[Atzmueller, 2007] Martin Atzmueller. *Knowledge-Intensive Subgroup Mining – Techniques for Automatic and Interactive Discovery*, volume 307 of *Dissertations in Artificial Intelligence-Infix*. IOS Press, March 2007.

[Aumann and Lindell, 2003] Yonatan Aumann and Yehuda Lindell. A Statistical Theory for Quantitative Association Rules. *Journal of Intelligent Information Systems*, 20(3):255–283, 2003.

[Grosskreutz *et al.*, 2008] Henrik Grosskreutz, Stefan Rüping, Nuhad Shaabani, and Stefan Wrobel. Optimistic estimate pruning strategies for fast exhaustive subgroup discovery. TR Fraunhofer IAIS, 2008.

[Han *et al.*, 2000] Jiawei Han, Jian Pei, and Yiwen Yin. Mining Frequent Patterns Without Candidate Generation. In Weidong Chen, Jeffrey Naughton, and Philip A. Bernstein, editors, *ACM SIGMOD Intl. Conf. on Management of Data*, pages 1–12. ACM Press, 2000.

[Jorge *et al.*, 2006] Alipio M. Jorge, Fernando Pereira, and Paulo J. Azevedo. Visual Interactive Subgroup Discovery with Numerical Properties of Interest. In *Proc. 9th Intl. Conf. on Discovery Science (DS 2006)*, pages 301–305, Berlin, October 2006. Springer.

[Klösgen, 1996] Willi Klösgen. Explora: A Multipattern and Multistrategy Discovery Assistant. In *Advances in Knowledge Discovery and Data Mining*, pages 249–271. AAAI Press, 1996.

[Lavrac *et al.*, 2004] Nada Lavrac, Branko Kavsek, Peter Flach, and Ljupco Todorovski. Subgroup Discovery with CN2-SD. *Journal of Machine Learning Research*, 5:153–188, 2004.

[Newman *et al.*,] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz. UCI Machine Learning Repository, 1998, http://www.ics.uci.edu/∼mlearn/mlrepository.html.

[Spenke, 2001] Michael Spenke. Visualization and Interactive Analysis of Blood Parameters with InfoZoom. *Artificial Intelligence In Medicine*, 22(2):159–172, 2001.

[Wrobel, 1997] Stefan Wrobel. An Algorithm for Multi-Relational Discovery of Subgroups. In *Proc. 1st Europ. Symp. on Principles of Data Mining and Knowledge Discovery (PKDD-97)*, pages 78–87, Berlin, 1997. Springer.

Learning a Metric during Hierarchical Clustering based on Constraints

Korinna Bade and Andreas Nürnberger
Otto-von-Guericke-University Magdeburg, Faculty of Computer Science, D-39106, Magdeburg, Germany
{korinna.bade,andreas.nuernberger}@ovgu.de

Abstract

Constrained clustering has many useful applications. In this paper, we consider applications, in which a hierarchical target structure is preferable. Therefore, we constrain a hierarchical agglomerative clustering through the use of MLB constraints, which provide information about hierarchical relations between objects. We propose an algorithm that learns a suitable metric according to the constraint set by modifying the metric whenever the clustering process violates a constraint. We furthermore combine this approach with an instance-based constrained clustering to further improve the cluster quality. Both approaches have proven to be very successful in a semi-supervised setting, in which constraints do not cover all existing clusters.

1 Introduction

Lately, a lot of work on constraint-based clustering has been published, e.g., [Bilenko *et al.*, 2004; Davidson *et al.*, 2007; Wagstaff *et al.*, 2001; Xing *et al.*, 2003]. However, all these works aim at deriving a single flat cluster partition, even though they might use a hierarchical cluster algorithm. In contrast to them, we are interested in obtaining a hierarchical structure of nested clusters [Bade and Nürnberger, 2008; Bade and Nürnberger, 2006]. This poses different requirements on the clustering algorithm.

There are many applications, in which a hierarchical cluster structure is more useful than a single flat partition. One such example is the clustering of text documents into a (personal) topic hierarchy. Such topics are naturally structured hierarchically. Furthermore, hierarchies can improve the access to the data for a user, if a large number of specific clusters is present, because the user can locate interesting topics step by step by several specializations.

After introducing our hierarchical setting, we show how constraints can be used in the scope of hierarchical clustering (Sect. 2). In Section 3, we review related work on constrained clustering in more detail. We then present an approach for hierarchical constrained clustering in Section 4 that learns a suitable metric during clustering, guided by the available constraints. The approach is evaluated in Section 5 with different hierarchical datasets of text documents.

2 Hierarchical Constrained Clustering

To avoid confusion with other approaches of constrained clustering as well as different opinions about the concept of hierarchical clustering, we define in this section the current problem from our perspective. Furthermore, we clarify the use of constraints in this setting.

Our task at hand is a semi-supervised hierarchical learning problem. The goal of the clustering is to uncover a hierarchical cluster structure H that consists of a set of clusters C and a set of hierarchically related pairs of clusters from C: $R_H = \{(c_1, c_2) \in C \times C \mid c_1 \geq_H c_2\}$ ($c_1 \geq_H c_2$ means that c_1 contains c_2 as a subcluster). Thus, the combination of C and R_H represents the hierarchical tree structure of clusters $H = (C, R_H)$. The data objects in O uniquely belong to one cluster in C. It is important to note that we specifically allow the assignment of objects to intermediate levels of the hierarchy. Such an assignment is useful in many circumstances as a specific leaf cluster might not exist for certain instances. As an example consider a document clustering task and a document giving an overview over a certain topic. As there might be several documents only describing a certain part of the topic and therefore forming several specific subclusters, the document itself naturally fits into the broader cluster as the scope of its content is also broad. This makes the whole problem a true hierarchical problem.

Semi-supervision is achieved through the use of must-link-before (MLB) constraints as introduced by [Bade and Nürnberger, 2008]. Other than the common pairwise must-link and cannot-link constraints [Wagstaff *et al.*, 2001], these constraints provide a hierarchical relation between different objects. Here, we use the triple representation:

$$MLB_{xyz} = (o_x, o_y, o_z). \quad (1)$$

In specific, this means that the items o_x and o_y should be linked on a lower hierarchy level than the items o_x and o_z as shown in Figure 1. This implies that o_y is contained in any cluster of the hierarchy that contains o_x and o_z. Furthermore, there is at least one cluster, which only contains o_x and o_y but not o_z.

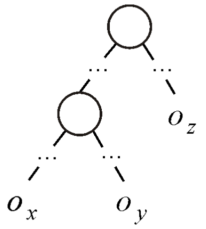


Figure 1: A MLB constraint in the hierarchy

MLB constraints can easily be extracted from a given hierarchy as shown in Figure 2. All item triples, which are linked through the hierarchy as shown in Figure 1 are selected. The example in Figure 2 shows two out of all possible constraints derived from the small hierarchy on the left. A given (known) hierarchy like this usually is a part of the overall hierarchy describing the data, i.e., $H_k = (C_k, R_{Hk})$ with $C_k \subseteq C$ and $R_{Hk} \subseteq R_H$ with some few data objects O_k assigned to these known clusters. This can, e.g., be data organized by a user in the past. In such an application, C_k is usually smaller than C, because a user will never have covered all available topics in his stored history. With the constraints generated from H_k, the clustering algorithm is constrained to produce a hierarchical clustering that preserves the existing hierarchical relations, while discovering further clusters and extracting their relations to each other and to the clusters in C_k, i.e., the constrained algorithm is supposed to refine the given structure by further data (see Fig. 3).

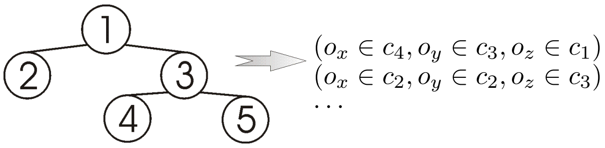

1
2
3
4
5
$(o_x \in c_4, o_y \in c_3, o_z \in c_1)$
$(o_x \in c_2, o_y \in c_2, o_z \in c_3)$
$\cdots$

Figure 2: Example of a MLB Constraint Extraction

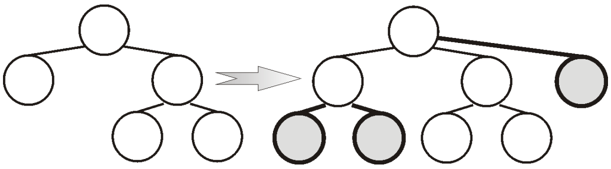


Figure 3: Hierarchy refinement/extension

3 Related Work

Constrained clustering covers a wide range of topics. The main focus of this section is on the use of triple and pairwise constraints. Besides these, other knowledge on the preferred clustering might be available, like cluster size, shape or number. The largest amount of existing work targets at a flat cluster partition and uses pairwise constraints as initially introduced by [Wagstaff *et al.*, 2001]. Cluster hierarchies are derived in [Bade and Nürnberger, 2006] and [Bade and Nürnberger, 2008].

[Bade and Nürnberger, 2006] proposed a first approach to learn a metric for hierarchical clustering. This idea is further elaborated by the work in this paper and will be described in detail in the following sections. [Bade and Nürnberger, 2008] describe two different approaches for constrained hierarchical clustering. The first one learns a metric based on the constraints previously to clustering. MLB constraints are therefor interpreted as a relation between similarities. A similar idea of using relative comparison is also described by [Schultz and Joachims, 2004], although it does not specifically target the creation of a cluster hierarchy. In this work, a support vector machine was used to learn the metric. [Bade and Nürnberger, 2008] also describe a second approach, in which MLB constraints are used directly during clustering without metric learning. This instance-based approach enforces the constraints in each cluster merge.

Like for hierarchical constrained clustering, approaches based on pairwise constraints can be divided in two types of approaches, i.e., *instance-based* and *metric-based* approaches. Existing instance-based approaches directly use constraints in several different ways, e.g., by using the constraints for initialization (e.g., in [Kim and Lee, 2002]), by enforcing them during the clustering (e.g., in [Wagstaff *et al.*, 2001]), or by integrating them in the cluster objective function (e.g., in [Basu *et al.*, 2004a]).

The *metric-based* approaches try to learn a distance metric or similarity measure that reflects the given constraints. This metric is then used during the clustering process (e.g., in [Bar-Hillel *et al.*, 2005], [Xing *et al.*, 2003], [Finley and Joachims, 2005], [Stober and Nürnberger, 2008]). The basic idea of most of these approaches is to weight features differently, depending on their importance for the distance computation. While the metric is usually learned in advance using only the given constraints, the approach in [Bilenko *et al.*, 2004] adapts the distance metric during clustering. Such an approach allows for the integration of knowledge from unlabeled objects and is also followed in this paper.

All the work cited in the two previous paragraphs targets on deriving a flat cluster structure through the use of pairwise constraints. Therefore, it cannot be compared to our work. However, we compare the newly proposed approach with the instance-based approach iHAC described by [Bade and Nürnberger, 2008]. Additionally, we combine our proposed metric learning with iHAC to create a new method considering both ideas. Gradient descent is used for metric learning and hierarchical agglomerative clustering (HAC) (with average linkage) for clustering. This choice was particularly motivated by the fact that the number of clusters (on each hierarchy level) is unknown in advance. Furthermore, we are interested in hierarchical cluster structures, which also makes it difficult to compare our results to the approaches based on pairwise constraints, because these algorithms were always evaluated on data with a flat cluster partitioning. An alternative would be a top-down application of flat partitioning algorithms. However, this would require the use of sophisticated techniques that estimate the number of clusters and are capable of leaving elements out of the partition (as done for noise detection).

4 Biased Hierarchical Agglomerative Clustering (biHAC)

In this section, we describe our biased hierarchical agglomerative clustering approach that learns a similarity measure based on violations of constraints by the clustering.

4.1 Similarity Measure

We decided to learn a parameterized cosine similarity, because the cosine similarity is widely used for clustering text documents, on which we focused in our experiments. Please note that the cosine similarity requires a vector representation of the objects (e.g., documents). Nevertheless, the general approach of MLB constraints and also our clustering method can in principle work with any kind of similarity measure and object representation. However, this requires modifications in the learning algorithm that fit to this different representation.

The cosine similarity can be parameterized with a symmetric, positive semi-definite matrix as also done by [Basu *et al.*, 2004b]:

$$\mathrm{sim}(o_i, o_j, W) = \frac{\vec{o}_i^T W \vec{o}_j}{|\vec{o}_i|_W |\vec{o}_j|_W} \tag{2}$$

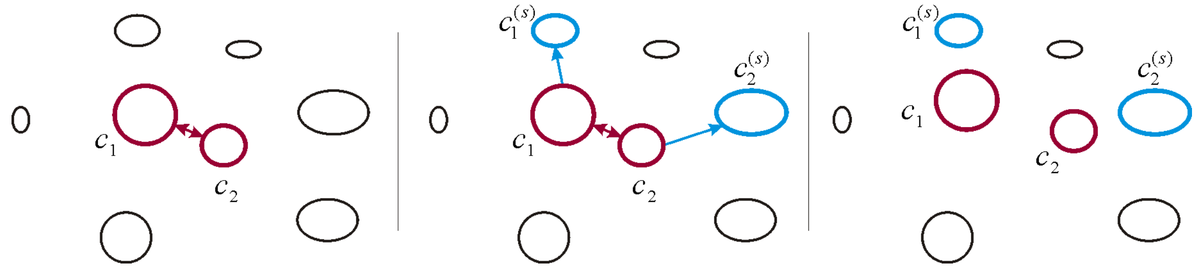


Figure 4: Weight adaptation in biHAC (left: the two closest clusters violate a constraint when merged; center: for each of the clusters the closest cluster not violating a constraint when merged is determined; right: similarity is changed to better reflect the constraints)

with $|\vec{o}|_W = \sqrt{\vec{o}^T W \vec{o}}$ is the weighted vector norm. Here, we use a diagonal matrix, which will be represented by the weight vector $\vec{w}$ including the diagonal elements of W. Such a weighting corresponds to an individual weighting of the individual dimensions (i.e., for each feature/term). Its goal is to express that certain features are more important for the similarity than others (of course according to the constraints). This is easy to interpret for a human, in contrast to a modification with a complete matrix. For the individual weights w_i in $\vec{w}$, it is required that they are greater than or equal to zero. Furthermore, we fixed their sum to the number of features to avoid extreme case solutions:

$$\forall w_i : w_i \geq 0 \quad \wedge \quad \sum_i w_i = n \tag{3}$$

Setting all weights to one yields the weighting scheme for the standard case of no feature weighting. This weighting scheme is valid according to (3).

4.2 Weight Learning

The basic idea of biHAC is to integrate weight learning in the cluster merging step of HAC. If merging the two most similar clusters c_1 and c_2 violates a constraint, this means that the similarity measure needs to be altered. A constraint MLB_{xyz} is violated by a cluster merge, if one cluster contains o_z and the other contains o_x but not o_y. Hence, the merged cluster would contain o_x and o_z but not o_y, which is not allowed according to MLB_{xyz}.

If such a violation is detected, the similarity should be modified so that c_1 and c_2 become more dissimilar. Furthermore, it is useful to guide the adaptation process to support a correct cluster merge instead. This can be achieved by identifying for both clusters a cluster ($c_1^{(s)}$ and $c_2^{(s)}$, respectively) with which they should have been merged instead, i.e., to which they should be more similar. These clusters shall not violate any MLB constraint when merged to the corresponding cluster of the two. Out of all such clusters, the one closest is selected as this requires the fewest weight changes. For clarification, an example is given in Fig. 4.

Please note that the items in $c_1^{(s)}$ and $c_2^{(s)}$ need not to be part of any constraint. For this reason, ”unlabeled” data can take part in the weight learning. Please note further that it is also not a good idea to change the similarity measure, if the clusters themselves already violate constraints due to earlier merges. In this case, the cluster does not reflect a single class. Therefore, weight adaptation is only computed, if the current cluster merge violated the constraints for the first time concerning the two clusters at hand.

With the four clusters determined for adaptation, two constraint triples on clusters rather than items can be built: $(c_1, c_1^{(s)}, c_2)$ and $(c_2, c_2^{(s)}, c_1)$. These express the desired properties just described. For learning, the similarity relation induced by such a constraint is most important:

$$(c_x, c_y, c_z) \rightarrow \mathrm{sim}(c_x, c_y) > \mathrm{sim}(c_x, c_z). \tag{4}$$

If this similarity relation is true, the correct clusters will be merged before the wrong ones, because HAC clusters the most similar clusters first. Based on (4), we can perform weight adaptation by gradient descent, which tries to minimize the error, i.e., the number of constraint violations. This can be achieved by maximizing

$$obj_{xyz} = \mathrm{sim}(c_x, c_y, \vec{w}) - \mathrm{sim}(c_x, c_z, \vec{w}) \tag{5}$$

for each (violated) constraint. This leads to

$$w_i \leftarrow w_i + \eta \Delta w_i = w_i + \eta \frac{\partial obj_{xyz}}{\partial w_i}, \tag{6}$$

for weight update with η being the learning rate defining the step width of each adaptation step.

To compute the similarity between the clusters, several options are possible (like in the HAC method itself). Here, we use the similarity between the centroid vectors $\vec{c}$, which are natural representatives of clusters. This has the advantage that it combines the information from all documents in the cluster. Through this, all data (including the data not occurring in any constraint) can participate in the weight update. Furthermore, the adaptation reflects the cluster level on which it occurred. Using cluster centroids, the final computation of Δw_i after differentiation is:

$$\begin{aligned} \Delta w_i &= \bar{c}_{x,i}(\bar{c}_{y,i} - \bar{c}_{z,i}) - \frac{1}{2}\mathrm{sim}(c_x, c_y, \vec{w})(\bar{c}_{x,i}^2 + \bar{c}_{y,i}^2) \\ &\quad + \frac{1}{2}\mathrm{sim}(c_x, c_z, \vec{w})(\bar{c}_{x,i}^2 + \bar{c}_{z,i}^2) \end{aligned} \tag{7}$$

with $\bar{c}_{x,i} = c_{x,i}/|\vec{c}_x|_{\vec{w}}$ and $c_{x,i}$ being the i-th component of $\vec{c}_x$. After all weights have been updated for one violated constraint by (6), all weights are checked and modified, if necessary, to fit our conditions in (3). This means that all negative weights are set to 0. After that, the weights are normalized to sum up to n.

To learn an appropriate weighting scheme, the clustering is re-run several times until the weights converge. The quality of the current weighting scheme can be assessed directly after each iteration based on the clustering result. This is done by computing the asymmetric H-Correlation [Bade and Benz, 2009] based on the given MLB constraints (see Section 5 for details). Based on this, we can determine the cluster error ce as one minus the H-Correlation. The cluster error can be used as a stopping criterion. If the cluster error does no longer decrease (for a specific number of

```
biHAC(documents $D$, constraints $MLB$, runs with no improvement $r_{ni}$, maximum runs $r_{max}$)
  Initialize $w$: $\forall i : w_i := 1$; best weighting scheme: $w^{(b)} := null$; best dendrogram: $DG^{(b)} := null$; best error $be := \infty$
  repeat
    Initialize clustering with $D$ and current similarity measure based on $w$
    Set current weighting scheme: $w^{(c)} := w$
    while not all clusters are merged do
      Merge the two closest clusters $c_1$ and $c_2$
      if merging $c_1$ and $c_2$ violates $MLB$ and the generation of $c_1$ and $c_2$ did not violate $MLB$ so far then
        Determine the most similar clusters $c_1^{(s)}$ and $c_2^{(s)}$ to $c_1$ and $c_2$, respectively, that can be merged according to $MLB$
        if $c_1^{(s)}$ exists then
          Adapt $w$ according to triple $(c_1, c_1^{(s)}, c_2)$
        end if
        if $c_2^{(s)}$ exists then
          Adapt $w$ according to triple $(c_2, c_2^{(s)}, c_1)$
        end if
      end if
    end while
    Determine cluster error on training data $ce$
    if $ce < be$ then
      $be := ce$; $w^{(b)} := w^{(c)}$;
      $DG^{(b)} :=$ current cluster solution
    end if
  until $w = w^{(c)}$ or $be$ did not improve for $r_{ni}$ runs or $r_{max}$ was reached
  return $DG^{(b)}$, $w^{(b)}$
```

Figure 5: The biHAC algorithm

clustering runs), the weight learning is terminated. Furthermore, it can be used to pick the best weighting scheme and dendrogram from all the ones produced during learning. Although the cluster error on the given constraints is optimistically biased because the same constraints were used for learning, it can be supposed that a solution with a good training error also has a reasonable overall error. The alternative is to use a hold-out set, which is a part of the constraint set but not used for weight learning, and estimate performance on this set instead. However, as constraints are often rare, it is usually not feasible to do so. The complete biHAC approach is summarized in Fig. 5.

As already shown by [Bade and Nürnberger, 2008], instance-based use of constraints can influence the clustering differently. We therefore combine the biHAC approach presented here with the iHAC approach presented by [Bade and Nürnberger, 2008]. After the weights are learned with the biHAC method, we add an additional clustering run based on the iHAC algorithm using the learned metric for similarity computation. This yields the biiHAC method.

4.3 Discussion

Before presenting the results of the evaluation, we want to address some potential problems and their solution through biHAC. First, we consider the scenario of unevenly scattered constraints. This means that constraints do not cover all existing clusters but a subset thereof. For our hierarchical setting, this means in specific that constraints are generated from labeled data of a subhierarchy. Unfortunately, most of the available literature on constrained clustering ignores this although it is the more realistic scenario and assumes evenly distributed constraints for evaluation. However, if constraints are unevenly distributed, a strong bias towards known clusters might decrease the performance for unknown clusters. This is especially true, if only the constraint set is considered during learning. In biHAC, we hope to circumvent or at least reduce this issue, because all data is used in the weight update. Therefore, the unlabeled data integrates knowledge about unknown clusters into the learning process. Furthermore, weight learning is problem oriented in biHAC (i.e., weights are only changed, if a constraint is violated during clustering), which might prevent unnecessary changes biased on the known clusters.

A second problem is connected to run-time performance with increasing number of constraints. If constraints are extracted based on labeled data, their number increases exponentially with increasing number of labeled data. As an example, consider the hierarchy of the Reuters 1 dataset used in the clustering experiments (cf. Sec. 5). Five labeled items per class generate 37200 constraints, which increases to 307800 constraints in the case of ten labeled items per class. In the first clustering run (with an initial standard weighting), biHAC only needs to compute about 23 weight adaptations in the first case and 36 adaptations in the second case. Thus, the exponential increase of constraints is not problematic for biHAC. Such a low number of weight adaptations is possible because biHAC focuses on the informative constraints, i.e., the constraints violated by the clustering procedure. Furthermore, weights are adapted for whole clusters. A detected violation, therefore, probably combines several constraint violations. However, only a single weight update is required for all of these.

Finally, we discuss the problem of contradicting constraints. We, hereby, do not mean inconsistency in the given constraint set but rather contradiction that occurs due to different hierarchy levels. As an example, consider two classes that have a common parent class in the hierarchy. A few features are crucial to discriminate the two classes. On the specific hierarchy level of these two classes, these features are boosted to allow for distinguishing both classes. However, on the more general level of the parent class, these features get reduced in impact, because both classes are recognized as one that shall be distinguished from others on this level. Given a certain hierarchy, there is an imbalance in the distribution of these different types with many more constraints describing higher hierarchy levels. This could potentially lead to an underrepresentation of the distinction between the most specific classes in the hierarchy. However, biHAC compensates this through its adaptation process based on violated cluster merges. This rather leads to a stronger focus on deeper hierarchy levels. There are two reasons for this. First, there are fewer clusters on higher hierarchy levels. And second, it is much more likely that a more specific cluster, which also contains fewer items, does not contain constraint violations from earlier merges, which forbids adaptation. Furthermore, cluster merges on higher levels combine several constraints at once, as indicated before.

5 Evaluation

We compared the biHAC approach, the iHAC approach [Bade and Nürnberger, 2008], and their combination (called biiHAC) for its suitability to our learning task. Furthermore, we used the standard HAC approach as a baseline. In the following, we first describe the used datasets and evaluation measures. Then we show and discuss the obtained results.

5.1 Datasets

As the goal is to evaluate hierarchical clustering, we used three hierarchical datasets. As we are particularly interested in text documents, we used the publically available banksearch dataset[1] and the Reuters corpus volume 1[2]. From these datasets, we generated three smaller subsets, which are shown in Fig. 6–8. The figures show the class structure as well as the number of documents directly assigned to each class. The first dataset uses the complete structure of the banksearch dataset but only the first 100 documents per class. For the Reuters 1 dataset, we selected some classes and subclasses that seemed to be rather distinguishable. In contrast to this, the Reuters 2 dataset contains classes that are more alike. We randomly sampled a maximum of 100 documents per class, while a lower number in the final dataset means that only less than 100 documents were available in the dataset.

- **Finance** (0)
  - Commercial Banks (100)
  - Building Societies (100)
  - Insurance Agencies (100)
- **Science** (0)
  - Astronomy (100)
  - Biology (100)
- **Programming** (0)
  - C/C++ (100)
  - Java (100)
  - Visual Basic (100)
- **Sport** (100)
  - Soccer (100)
  - Motor Racing (100)

Figure 6: Banksearch dataset

- **Corporate/Industrial** (100)
  - Strategy/Plans (100)
  - Research/Development (100)
  - Advertising/Promotion (100)
- **Economics** (59)
  - Economic Performance (100)
  - Government Borrowing (100)
- **Government/ Social** (100)
  - Disasters and Accidents (100)
  - Health (100)
  - Weather (100)

Figure 7: Reuters 1 dataset

- **Equity Markets** (100)
- **Bond Markets** (100)
- **Money Markets** (100)
  - Interbank Markets (100)
  - Forex Markets (100)
- **Commodity Markets** (100)
  - Soft Commodities (100)
  - Metals Trading (100)
  - Energy Markets (100)

Figure 8: Reuters 2 dataset

All documents were represented with $tf \times idf$ document vectors. We performed a feature selection, removing all terms that occurred less than 5 times, were less than 3 characters long, or contained numbers. From the rest, we selected 5000 terms in an unsupervised manner as described in [Borgelt and Nürnberger, 2004]. To determine this number we conducted a preliminary evaluation. It showed that this number still has a small impact on initial clustering performance, while a larger reduction of the feature space leads to decreasing performance.

We generated constraints by using labeled data as described at the end of Section 2. For each considered setup, we have randomly chosen five different samples. However, the same labeled data is used for all algorithms to allow a fair comparison. We created different settings reflecting different distributions of constraints. Setting (1) uses labeled data from all classes and therefore equally distributed constraints. Two more settings with unequally distributed constraints were evaluated by not picking labeled data from a single leaf node class (setting (2)) or a whole subtree (setting (3)). Furthermore, we used different numbers of labeled data given per class. We specifically investigated small numbers of labeled data (with a maximum of 30) as we assume from an application oriented point of view that it is much more likely that labeled data is rare.

5.2 Evaluation Measures

We used two measures to evaluate and compare the performance of our algorithms. First, we used the F-score gained in accordance to the given dataset, which is supposed to be the true cluster structure that shall be recovered. For its computation in an unlabeled cluster tree (or in a dendrogram), we followed a common approach that selects for each class in the dataset the cluster gaining the highest F-score on it. This is done for all classes in the hierarchy. For a higher level class, all documents contained in subclasses are also counted as belonging to this class. Please note that this simple procedure might select clusters inconsistent with the hierarchy or multiple times in the case of noisy clusters. Determining the optimal and hierarchy consistent selection has a much higher time complexity. However, the results of the simple procedure are usually sufficient for evaluation purposes and little is gained from enforcing hierarchy consistency. We only computed the F-score on the unlabeled data, as we want to measure the gain on the new data. As F-score is a class specific value, we computed two mean values: one over all leaf node classes and one over all higher level classes.

Applying the F-score as described potentially leads to an evaluation of only a part of the clustering, because it just considers the best cluster per class (even though that might be the most interesting part). Therefore, we furthermore used the asymmetric H-Correlation [Bade and Benz, 2009], which can compare entire hierarchies. In principle, it measures the (weighted) fraction of MLB constraints that can be generated from the given dataset and are also found in the learned dendrogram. It is defined as:

$$H_a = \frac{\sum_{\tau \in MLB_l \cap MLB_g} w_g(\tau)}{\sum_{\tau \in MLB_g} w_g(\tau)}, \tag{8}$$

whereby MLB_l is the constraint set generated from the dendrogram, MLB_g is the constraint set generated from the dataset, and $w_g(\tau)$ weights the individual constraints. We used the weighting function described by [Bade and Benz, 2009] that gives equal weight to all hierarchy nodes in the final result.

5.3 Results

In this section, we present the results obtained in our experiments. The following parameter sets were used for biHAC: We used a constant learning rate of five and limited the number of iterations to 50 to limit the necessary

[1] Available for download at the StatLib website (http://lib.stat.cmu.edu); Described in [Sinka and Corne, 2002]

[2] Available from the Reuters website (http://about.reuters.com/researchandstandards/corpus/)

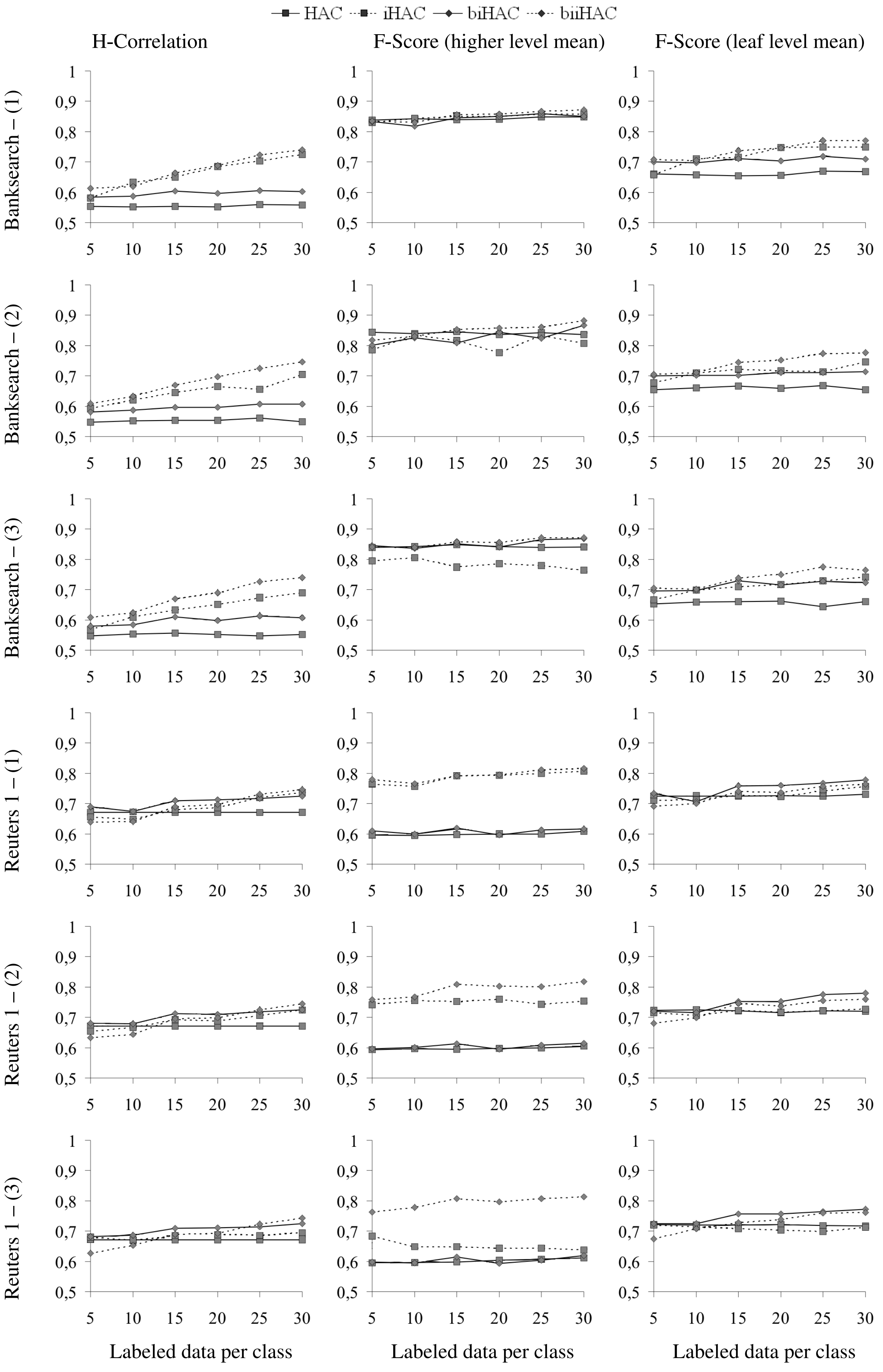


Figure 9: Results for the Banksearch and the Reuters 1 data set

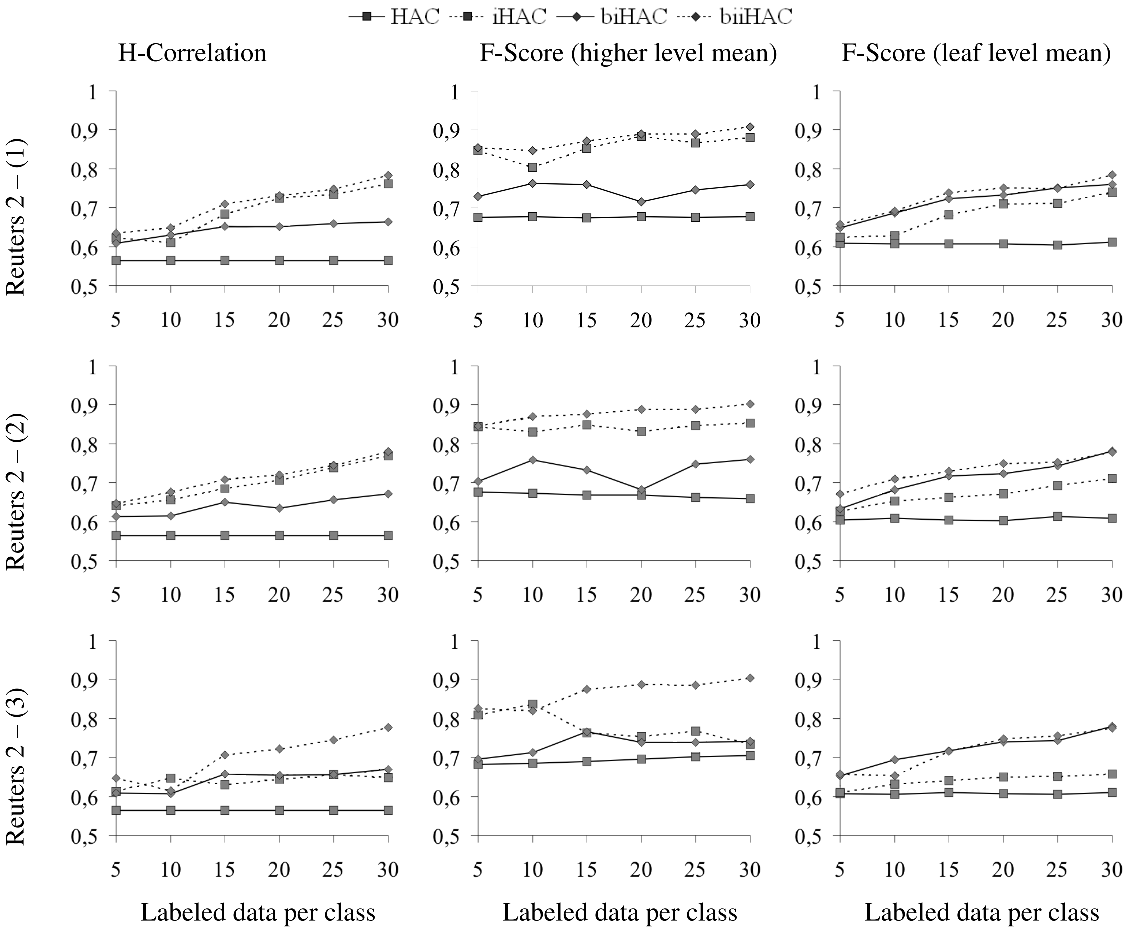


Figure 10: Results for the Reuters 2 data set

run-time of our experiments. In preliminary experiments, these values showed to behave well.

Figures 9 and 10 show our results on the three datasets. Each column of the two figures corresponds to one evaluation measure. Each dataset is represented with 3 diagram rows, the first showing the results with labeled data from all classes (setting (1)), the second showing results with one leaf node class unknown (setting (2)), and the third showing results with a complete subtree unknown (setting (3)). Each diagram shows the performance of the algorithms on the respective measure with increasing number of labeled data per class.

In general, it can be seen that all three approaches are capable of improving cluster quality through the use of constraints. However, the specific results differ over the different settings and datasets analyzed. In our discussion, we start with the most informed setting, i.e., (1), before we turn to settings (2) and (3).

For the Banksearch dataset (row 1 in Fig. 9), the overall cluster tree improvement as measured with the H-Correlation is constantly increasing with increasing number of constraints for iHAC and biiHAC, while biHAC itself rather adds a constant improvement to the baseline performance of HAC. biiHAC is hereby mostly determined by iHAC as it has about equal performance. Similar behavior can be found for the F-score measure on the leaf level. On the higher level, almost no improvement is achieved with any method, probably because the baseline performance is already quite good.

For the Reuters 1 dataset (row 4 in Fig. 9), iHAC and biiHAC have a smaller overall improvement in comparison to the Banksearch dataset. However, both methods highly improve the higher level clusters. This can be explained by the nature of the dataset, which contains mostly very distinct classes. Therefore, weighting can improve little. Similarities for the higher levels are hard to find. The used features based on term occurrences are not expressive enough to recover the structure in this dataset. Here, the instance-based component can clearly help, because it does not require similarities in the representation of the documents.

For the Reuters 2 dataset (row 1 in Fig. 10), all algorithms are again capable of largely increasing the H-Correlation. While on the higher levels, the instance-based component has again advantages, biHAC can better adapt to the leaf level classes. The combination of both algorithms through biiHAC again succeeds in producing an overall good result towards the better of the two methods on the specific levels. This dataset benefits most from integrating constraints, which can also be explained by its nature. It contains many very similar documents. Therefore, feature weights can reduce the importance of terms very frequent over the whole dataset and boost important words on all hierarchy levels.

Summing up, the combined approach biiHAC provides good results on all datasets in the case of evenly distributed constraints. There is no clear winner between the instance-based and the metric-based approach. Specific performance gains depend on the properties of the dataset. Most importantly, the chosen features need to be expressive enough to explain the class structure in the dataset, if a metric shall be learned successfully.

Next, we analyze the behavior in the case of unevenly distributed constraints in settings (2) and (3). In general, the performance gain is less, if more classes are unknown in advance. However, this is also expected, as this means a lower number of constraints. Nevertheless, analyzing class specific values (which are not printed here) shows that classes with no labeled data usually still increase in performance in the biHAC approach. Thus, knowledge about some classes not only helps in distinguishing between these classes but also reduces the confusion between known and unknown classes. For iHAC, this is a lot more problematic because it completely ignores the possibility of new classes. A single noisy clustered instance that is part of a constraint can therefore result in a complete wrong clustering of an unknown class, which cannot provide constraints for itself to ensure its correct clustering.

Let us look in the results in detail. For the Banksearch data (rows 1–3 in Fig. 9), biHAC and biiHAC have a stable performance with increasing number of unknown classes, while iHAC looses performance. On the higher level, its performances even deteriorates under the baseline performance of HAC.

For the Reuters 1 data (rows 4–6 in Fig. 9), the main difference can be found on the higher cluster levels. iHAC looses its good performance on the higher level although it is still better than the baseline. In setting (3), it also shows that more constraints are even worse for its performance. On the other hand, biiHAC can keep up its good performance. Although the metric alone does not effect the higher level performance, it is sufficient to ensure that the instance-based component does not falsely merge clusters, especially if they correspond to unknown classes.

For the Reuters 2 data (Fig. 10), again the iHAC approach looses performance with increasing number of unknown classes, while biHAC as well as biiHAC can keep up their good quality. The effect is the same as for the Reuters 1 dataset.

Summing up over all datasets, the metric learning through biHAC is very resistant to an increasing number of unknown classes. This is probably due to the fact that it learns the metric during clustering and thereby incorporates data from the unknown classes in the weight learning. It is remarkably to note that its combination with an instance-based approach in biiHAC has often an even better performance. Although, instance-based constrained clustering itself is unsuitable, the more classes are unknown, biiHAC largely benefits further from the instance-based use of constraints.

6 Conclusion

In this paper, we introduced an approach for constrained hierarchical clustering that learns a metric during the clustering process. This approach as well as its combination with an instance-based approach presented earlier was shown to be very successful in improving cluster quality. This is especially true for semi-supervised settings, in which constraints do not cover all existing clusters. This is in contrast to the instance-based method alone, whose performance heavily suffers in this case.

References

[Bade and Benz, 2009] K. Bade and D. Benz. Evaluation strategies for learning algorithms of hierarchies. In *Proc. of the 32nd Annual Conference of the German Classification Society*, 2009.

[Bade and Nürnberger, 2006] K. Bade and A. Nürnberger. Personalized hierarchical clustering. In *Proc. of 2006 IEEE/WIC/ACM Int. Conf. on Web Intelligence*, 2006.

[Bade and Nürnberger, 2008] K. Bade and A. Nürnberger. Creating a cluster hierarchy under constraints of a partially known hierarchy. In *Proc. of the 2008 SIAM Int. Conference on Data Mining*, pages 13–24, 2008.

[Bar-Hillel *et al.*, 2005] Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6:937–965, 2005.

[Basu *et al.*, 2004a] S. Basu, A. Banerjee, and R. Mooney. Active semi-supervision for pairwise constrained clustering. In *Proc. of the 4th SIAM Int. Conf. on Data Mining*, 2004.

[Basu *et al.*, 2004b] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proc. of the 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2004.

[Bilenko *et al.*, 2004] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proc. of the 21st Int. Conf. on Machine Learning*, pages 81–88, 2004.

[Borgelt and Nürnberger, 2004] C. Borgelt and A. Nürnberger. Fast fuzzy clustering of web page collections. In *Proc. of PKDD Workshop on Statistical Approaches for Web Mining (SAWM)*, 2004.

[Davidson *et al.*, 2007] I. Davidson, S. S. Ravi, and M. Ester. Efficient incremental constrained clustering. In *Proc. of the 13th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining.*, 2007.

[Finley and Joachims, 2005] T. Finley and T. Joachims. Supervised clustering with support vector machines. In *Proc. of the 22nd Int. Conf. on Machine Learning*, 2005.

[Kim and Lee, 2002] H. Kim and S. Lee. An effective document clustering method using user-adaptable distance metrics. In *Proc. of the 2002 ACM symposium on Applied computing*, pages 16–20, 2002.

[Schultz and Joachims, 2004] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *Proc. of Neural Information Processing Systems*, 2004.

[Sinka and Corne, 2002] M. Sinka and D. Corne. A large benchmark dataset for web document clustering. In *Soft Computing Systems: Design, Management and Applications, Vol. 87 of Frontiers in Artificial Intelligence and Applications*, pages 881–890, 2002.

[Stober and Nürnberger, 2008] S. Stober and A. Nürnberger. User modelling for interactive user-adaptive collection structuring. In *Postproc. of 5th Int. Workshop on Adaptive Multimedia Retrieval*, 2008.

[Wagstaff *et al.*, 2001] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proc. of 18th Int. Conf. on Machine Learning*, pages 577–584, 2001.

[Xing *et al.*, 2003] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, pages 505–512. 2003.

Learning SQL for Database Intrusion Detection using Context-Sensitive Modelling
(Re-submission)

Christian Bockermann, Martin Apel, Michael Meier
Technische Universität Dortmund
44221 Dortmund, Germany
{christian.bockermann,martin.apel,michael.meier}@udo.edu

Abstract

Modern multi-tier application systems are generally based on high performance database systems in order to process and store business information. Containing valuable business information, these systems are highly interesting to attackers and special care needs to be taken to prevent any malicious access to this database layer. In this work we propose a novel approach for modelling SQL statements to apply machine learning techniques, such as clustering or outlier detection, in order to detect malicious behaviour at the database transaction level. The approach incorporates the parse tree structure of SQL queries as characteristic e.g. for correlating SQL queries with applications and distinguishing benign and malicious queries. We demonstrate the usefulness of our approach on real-world data.

1 Introduction

The majority of today's web-based applications does rely on high performance data storage for business processing. A lot of attacks on web-applications are aimed at injecting commands into database systems or try to otherwise trigger transactions to gain unprivileged access to records stored in these systems. See [1] for a list of popular attacks on web applications.

Traditional network-based firewall systems offer no protection against these attacks, as the malicious (fractions of) SQL or tampered requests are located at the application layer and thus are not visible to most of these systems.

The usual way of protecting modern application systems is by introducing detection models on the network layer or by the use of web application firewall systems. These systems often employ a misuse detection approach and try to detect attacks by matching network traffic or HTTP request against a list of known attack patterns. A very popular system based on pattern matching is for instance the Snort IDS [2]. Another project aiming at the detection of tampered HTTP requests is the ModSecurity module, which provides a rule-engine for employing pattern based rules within a Web-Server [3].

Instead of using pattern based approaches, there exists a variety of papers on employing anomaly-based methods for detecting web-based intrusions [4; 5; 6]. These either try to analyze log-files or protocol-level information to detect anomalies based on heuristics or data-mining techniques. We earlier proposed a rule based learning approach using the ModSecurity module in [7].

These approaches are rooted at the network or application protocol layer. In this work we focus on the detection at the database layer, i.e. the detection of anomalous SQL statements, that are either malicious in the sense that they include parts of injected code or differ from the set of queries usually issued within an application. The main contribution of our work is the use of a grammar based analysis, namely tree-kernel based learning, which became popular within the field of natural language processing (NLP). Our approach incorporates the parse tree structure of SQL queries as characteristic e.g. for correlating SQL queries with applications and distinguishing benign and malicious queries. By determining a context sensitive similarity measure we can locate the nearest legal query for an malicious statements which helps in root cause analysis.

The remainder of this paper is organized as follows: Section 2 states the problem in detail and gives an overview of related work regarding intrusion detection in databases. In Section 3 we give a short introduction to kernel-based learning algorithms in general and their application on structured data in detail. Following this overview we define our tree-kernel based method and describe its application to learning SQL for intrusion detection in databases in Section 4. Finally we present our results on real-world data in Section 5 and summarize our experiments.

2 Problem & Related Work

Executing malicious statements on a database may result in severe problems, which can range from exposure of sensitive information to loosing records or broken integrity. Once an attacker manages to inject code into a database this will likely not only affect specific records, but may lead to a compromise of the complete application environment. This in turn can cause severe outages with respect to data records and a company's public reputation.

Although the risk may seem low on a first glance, given the database layer is separated from the public interface (web/presentation layer) and not directly accessible from the outside, anomalous queries caused by e.g. SQL injection attacks are a widespread problem. The *Web Hacking Incident Database* provides a listing of recent web hacks, a lot of them relying on SQL injections [8].

There have been approaches to apply data-mining and machine learning methods to detect intrusions in databases. Lee et al [9] suggest learning fingerprints of access patterns of genuine database transactions (e.g. read/write sequences) and using them to identify potential intrusions. Typically there are many possible SQL queries, but most of them only differ in constants that represent the user's input. SQL queries are summarized in fingerprints (regu-

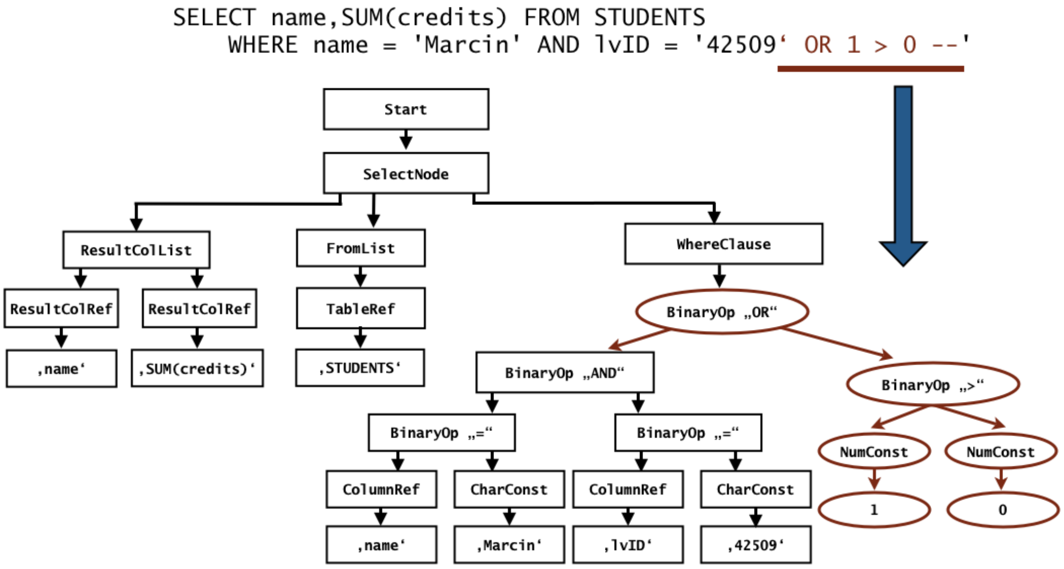


Figure 1: SQL parse tree of an SQL injection

lar expressions) by replacing the constants with variables or wild-cards. Such fingerprints capture some structure of the SQL queries. Following the approach of [9], queries not matching any of the existing fingerprints are reported as malicious. A drawback of this approach is its inability to correlate and identify fingerprints with applications.

In [10] the authors also try to detect SQL injections by a kind of fingerprints. They use parse trees of queries as fingerprints for the queries structure. The main idea here is to compare the parse tree of an SQL statement before and after user-variables have been inserted. Injected SQL fragments will typically significantly change the trees structure. An example of such structural changes in the parse tree of a query is shown in figure 1. In this figure, the rounded nodes of the tree indicate the additional parts that have been added due to the injection SQL fragment `' OR 1 > 0 --`. As this work only uses a one-to-one comparison on parse-trees it is missing any generalization capabilities and thus not applicable for machine learning methods, such as clustering and outlier detection.

A similar grammar-based approach has been used in [11], which studied the use of syntax-aware analysis of the FTP protocol using tree-kernel methods on protocol parse-trees. A slightly different approach was taken in [12] where the parse tokens are used along with their values to detect anomalies in HTTP-traffic. The latter approach does not use the full parse tree but its leaves. Our work is similar to [11; 12] in the sense that it employs machine learning methods on syntax trees derived from a protocol parser.

Also approaches on investigating data dependencies have been proposed in [13] and [14]. Data dependencies refer to access correlations among sensitive data items. Data dependencies are generated in form of classification rules like *before an update of item1 a read of item2 is likely*. Transactions not compliant to these rules are flagged as malicious. Srivastava et al [14] further distinguish different levels of sensitivity of data items which need to be specified by hand. Both approaches ignore the structure of SQL queries and are unable to correlate SQL queries with applications. A more recent work has been presented in [15], focusing on the sequential nature of SQL queries. These studies also make use of a smart modelling technique to easily apply data mining methods on their SQL representations.

3 A Grammar-based Modelling

Since most learning approaches work on vectorized data, a key issue when using machine learning for intrusion detection is the representation of monitored data to apply any learning algorithm. A popular technique in IDS is the exhaustive creation of n-grams, yielding histogram vectors for observed input data. These do not maintain any syntactical information of SQL. A little more syntax is regarded by creating *term-vectors* of a query. A *term-vector* can be obtained by splitting the query in a “proper way”, i.e. by splitting on whitespace characters (optionally maintaining quoted strings).

As in this work we are dealing with the detection of malicious database queries, we choose a grammar based approach to represent SQL queries. We propose two alternative modelling approaches for making SQL queries suitable for machine learning.

3.1 Parsing SQL

The basic idea of [10] is to detect SQL injection attacks by means of changes in a queries syntax tree. An example of such a tree has been shown before (see figure 1). In order to obtain such a parse tree, a parser for the SQL dialect is required. Usually complex parsers are automatically generated based on a given grammar description using tools such as *yacc*, *antlr* or *javacc*. Unfortunately, the availability of proper grammar descriptions for SQL is pretty sparse and most existing parser implementations are tightly wired into the corresponding DBMS, making it laborious to extract a standalone parser.

We therefore decided to modify an existing open-source DBMS, in our case the Apache Derby database, which provides a standalone deployment. The Derby parser is itself generated off a grammar file using *javacc*, but does not explicitly output a syntax tree suitable for our decomposition. Using the tree-interface of the parser, we derived a tree-inspection tool which traverses the tree object of a query and writes out the corresponding node information.

3.2 Vectorization of SQL Queries

To incorporate more syntax, we determine the parse tree of a query. As we are interested in the detection of abnormal queries within our database application, we are looking for a similarity measure for the space of structured objects, i.e. the space of valid SQL parse trees. Thus, we are faced with the problem of having to create a distance function for matching trees.

Definition: *Let q be an SQL query and τ_q the parse tree of q, identifying with τ_q the root node of the tree. Each node n within that tree is labeled with an identifier* $\mathrm{type}(n)$*, reflecting the node type.*

For a node n within τ_q we denote by $\mathrm{succ}(n)$ *the ordered set of successors of n and by* $\mathrm{succ}_i(n)$ *the ith child of n.*

This definition is basically just a formalization of a query's syntax tree. It allows us to enlist the production or grammar rules, which generate a given SQL query q. This list of production rules will be defined as follows:

Definition: *For a node n within the parse tree τ_q of a query q, the list of production rules $P(n)$ is given by*

$$P(n) = \biguplus_{c \in \mathrm{succ}(n)} \{\mathrm{type}(n) \rightarrow \mathrm{type}(c)\} \uplus \biguplus_{c \in \mathrm{succ}(n)} P(c).$$

Given $P(n)$, denote by $|P(n)|_r$ the number of times the rule r occurs in $P(n)$.

Please note that we use the $\uplus$ notation here for list concatenation, thus, the resulting list may contain the same rule more than once. Now, denoting with Q the set of all valid trees for a given SQL dialect, these simple definitions allow us to define a mapping $\varphi : Q \rightarrow \mathbb{R}^n$, by following the *bag of words* approach known from text classification tasks like *spam detection* as proposed in [16].

Definition: *Let R be the sorted set of all possible production rules, defined by some SQL grammar and r_i the ith rule of R. For an SQL query q with the associated parse tree τ_q the rule vector* $\mathbf{v} \in \mathbb{R}^{|R|}$ *is given by $v_i = |P(\tau_q)|_{r_i}$. The function φ maps an SQL query q to the vector space $\mathbb{R}^{|R|}$ by $\varphi(q) = \mathbf{v}$.*

Since an SQL query usually consists of only a small fraction of the complete SQL grammar, these rule vectors are typically very sparse. Based on this mapping we can now define a distance measure on SQL queries using any distance function Δ in the vector space $\mathbb{R}^{|R|}$ by defining the corresponding distance function Δ_{SQL} using

$$\Delta_{SQL}(q_1, q_2) := \Delta(\varphi(q_1), \varphi(q_2)), \tag{1}$$

where q_1, q_2 are any two SQL statements of a common dialect. This allows for the application of a wide range of distance based learning algorithms such as clustering or outlier detection.

4 Using Tree-Kernels for SQL Grammars

The simple vectorization of SQL queries defined above includes a weak context based reasoning to be used within the distance measure in $\mathbb{R}^{|R|}$. It can be seen as an an *explicit feature extraction* approach, as it explicitly creates feature vectors from SQL statements. Unfortunately, the rule counting does only incorporate direct antecessor relationships, limiting the contextual scope.

4.1 Introduction to Tree-Kernels

To overcome these limitations the natural language processing community makes use of context based tree-kernels, which provide a so-called *kernel-function* over trees. In the machine learning community kernel-based methods have received a lot of attention not ultimately owing to the well-known *support vector machine* method, which has also been used for intrusion detection [17; 12]. These methods make use of a kernel-function to measure the similarity between instances of some input space $\mathcal{X}$, i.e. a kernel k is symmetric and positive (semi-) definite function

$$k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$$

which implicitly computes an inner product in a reproducing kernel Hilbert space. There exists kernel functions for complex structures like trees or graphs, which are often defined as convolution kernels [18]. For these kernels one defines a kernel over atomic structures and defines the convolution kernel for complex objects by combining the kernel function of its sub structures.

In [19] Collins and Duffy propose a simple kernel over trees for use in natural language processing. The basic idea is to capture structural information over trees in the kernel function by incorporating all sub-trees occuring within the trees of interest. Let $\mathcal{T}$ be the space of all trees in question and denote with T the ordered set of all possible sub-trees in $\mathcal{T}$. For a tree $\tau \in \mathcal{T}$ denote by $h_i(\tau)$ the number of occurrences of the i-th sub-tree of $\mathcal{T}$ in τ and with $N(\tau)$ the set of all nodes in τ. For two trees τ_1, τ_2 the tree-kernel in [19] is defined by

$$K_C(\tau_1, \tau_2) = h_i(\tau_1)h_i(\tau_2) = \sum_{\substack{n_1 \in N(\tau_1),\\ n_2 \in N(\tau_2)}} \Delta(n_1, n_2).$$

The function Δ is defined as follows

$$\Delta(n_1, n_2) = \begin{cases} 0 & \text{if } P(n_1) \neq P(n_2) \\ \lambda & \text{if } H(n_1) = H(n_2) = 1 \\ \Delta^*(n_1, n_2) & \text{otherwise,} \end{cases}$$

where $H(n_i) = \mathrm{height}(n_i)$ and $\Delta^*(n_1, n_2)$ is recursively defined as

$$\Delta^*(n_1, n_2) = \lambda \prod_{k=1}^{|\mathrm{succ}(n_1)|} [1 + \Delta(\mathrm{succ}_k(n_1), \mathrm{succ}_k(n_2))]$$

Roughly speaking, this kernel measures the similarity of two trees by the set of common sub trees. As it does not consider the context of a sub tree, Zhou et al [20] designed a *context-sensitive convolution* tree-kernel, by taking into account a sub trees' context by means of its ancestors.

Starting with a tree τ, a root node path of length l in τ is a path from the root node τ or any of its successors to a node in τ, which has a length of l. Following the notation of [20], the set of all root node paths for a tree τ_j with a maximal length of m is denoted by $N^m[j]$. Given a maximum length m for the root node paths considered, the context-sensitive tree-kernel is given be

$$K_{CSC}(\tau_1, \tau_2) = \sum_{i=1}^{m} \sum_{\substack{n_1^i[1] \in N_1^i[1],\\ n_2^i[2] \in N_1^i[2]}} \Delta_{CSC}(n_1^i[1], n_2^i[2]),$$

where $n_1^i[j] = (n_1, n_2, \ldots, n_i)[j]$ denotes a root node path of length i in tree τ_j. This kernel will therefore incorporate the similarity of common sub-trees.

4.2 Using Tree-Kernels for SQL Parse-Trees

As mentioned in the beginning, the use of tree-kernels in intrusion detection has been proven to provide a syntax-oriented analysis in protocols such as FTP or HTTP [11; 12]. To exploit the benefit of syntax-level awareness in SQL query-analysis, we derive the distance measure induced by a tree-kernel function to directly measure the similarity of SQL queries by means of their parse-trees.

For a kernel k and examples x_1, x_2, such a distance can be obtained by

$$d(x_1, x_2) = \sqrt{k(x_1, x_2) - 2k(x_1, x_2) + k(x_1, x_2)}. \quad (2)$$

Using a tree-kernel we can therefore use this kernel to directly compute the distance of two SQL parse-trees using (2).

5 Experimental Analysis and Results

For an evaluation of the different modelling approaches we collected data of the popular Typo3 content management system. This application heavily depends on the use of SQL for various tasks beyond page content storage, such as session-persistence, user-management and even page-caching.

Model	Ratio 200:15			Ratio 1000:15		
	TPR	FPR	time	TPR	FPR	time
3-gram	0.667	0.000	71s	0.667	0.002	643s
4-gram	0.333	0.000	149s	0.733	0.002	1055s
Term vectors	0.667	0.005	2s	0.733	0.002	283s
SQL vectors	0.867	0.000	16s	0.867	0.001	67s

Table 1: Separation capabilities of the different models based on a 10-fold cross-validation.

We created a set of distinct queries and added synthetic attacks, which closely reflect modifications that would follow from SQL injections, by inserting typical injection vectors such as `OR 'a' = 'a'` or the like into legal statements. The intention was to observe whether, using different models, the SVM is to distinguish between legal and malicious statements even though the latter were only marginally different. We created two sets with different ratios of normal to malicious queries, one with 200:15, the other with 1000:15 queries.

5.1 Importance of Context

A central question in our work is the importance of contextual information when analyzing SQL queries. We therefore analyzed approaches such as n-grams, term-vector and the SQL vectorization described in section 3.2. In this experiment we did not mean to train a detector, but wanted to explore the expressiveness of the different models and determined the detection rate (TPR) and the false-positive rate (FPR) of the different modelling approaches. As learning algorithm we used an SVM approach within a 10-fold cross-validation.

As you can see from table 1, the use of context information results in performance gains especially with respect to the detection rate (TPR) and the fraction of false positives (FPR). This supports our thesis on the importance of the context when analyzing SQL queries. It is worth noting, that the variance in TPR/FPR within the 10-fold cross validation proved to be much smaller for the context-sensitive methods. Additionally, the training time using term- or sql-vectorization decreased due to the smaller number of (irrelevant) attributes. The times in table 1 refer to the complete parameter-optimization and 10-fold cross-validation process.

5.2 Query Analysis using Tree-Kernels

Using the tree-kernel similarity we are interested in analyzing an application's structure by means of different sets of similar statements used. Therefore we used the kernel similarity within an interval self-organizing map (ISOM) to create a visualization of an application's statements. In figure 2 you see the ISOM of 200 regular queries taken from Typo-3 (dots), supplemented by 15 modified "malicious" modifications (squares).

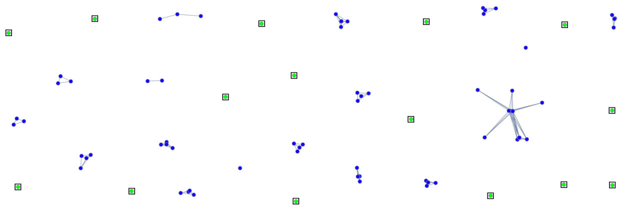


Figure 2: ISOM of 215 Typo-3 queries (200 legal, 15 anomalous), created by the CSC tree-kernel ($\lambda = 0.6, m = 10$).

As can be seen in figure 2 the kernel does consolidate similar queries into clusters, an inspection of the clustered regions revealed very reasonable groups, such as "all page-content queries", "all session update queries" and so on. The heaps of dots turned out to be of a very similar structure, only differing in terminal symbols. Further adding edges to the ISOM showed, that the modified queries are consolidated very late, showing that they are highly dissimilar.

5.3 Intra-Cluster ISOMs

As the ISOM experiments proved to be useful to get a feeling for the similarity measure, we employed a KMedoids clustering algorithm based on the tree-kernel distance and inspected the clusters by creating ISOMs of each cluster separately. Figure 3 shows the ISOM of a cluster containing "attacks" which are similar to the majority of the queries, but differ by injected SQL fragments.

Within this cluster the anomalous queries is the one most dissimilar from all other, resulting in isolation. The queries in the left-hand group are related to selecting language-specific content from the database, whereas the group on the right contains queries selecting page-content related to a user-id `UID`. The anomalous query contains an additional `OR UID > 0`, neutralizing the UID check.

This yields a two-way analysis which uses a clustering approach to first group the different kinds of statements and then uses an intra-cluster outlier detection for the detection of malicious queries.

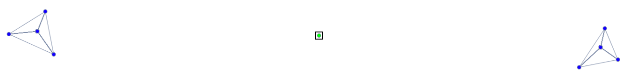


Figure 3: Intra-Cluster ISOM of a cluster consisting of 46 legal queries and one single anomalous modification.

6 Conclusions and Future Work

We presented two approaches for a context sensitive modelling/fingerprinting of SQL queries by use of generic models. Using tree-kernels for analyzing SQL statements brings together the results of natural language processing with a highly structured query language. The results confirm the benefit of incorporation of syntax information of previous works [11; 12] in the domain of SQL queries.

The consideration of the SQL structures shows performance gains in both performance and speed, the later due to the fewer but far more meaningful features. Compared to previous approaches the tree-kernels allow for a similarity measure on SQL statements providing flexible generalization capabilities.

However, a drawback in the use of tree-kernels is their computational overhead. Given a set of 1015 queries, the computation of the kernel matrix took about 210 seconds. Use of hierarchical models, such as hierarchical clustering, may lower the impact of this performance decrease for future detection models.

Here, our first Clustering and ISOM experiments in 5 show the usefulness of tree-kernels as a similarity measure in order to visualize SQL queries in applications. However, the tree-kernel approach still offers a lot of optimization possibilities and needs further investigation. In future works we therefore plan on using inter-cluster outlier detection to create hierarchical anomaly detection models based on tree-kernels over SQL parse-trees.

References

[1] Open Web Application Security Project. The Top list of most severe web application vulnerabilities, 2004.

[2] M. Roesch. Snort: Lightweight intrusion detection for networks. In *Proc. of LISA*, pages 229–238. USENIX, 1999.

[3] I. Ristic. ModSecurity - A Filter-Module for the Apache Webserver, 1998.

[4] C. Kruegel and G. Vigna. Anomaly Detection of Web-based Attacks. In *Proc. of ACM CCS*, pages 251–261. ACM Press, 2003.

[5] C. Kruegel, G. Vigna, and W. Robertson. A Multi-model Approach to the Detection of Web-based Attacks. *Computer Networks*, 48(5):717–738, 2005.

[6] F. Valeur, G. Vigna, C. Kruegel, and E. Kirda. An Anomaly-driven Reverse Proxy for Web Applications. In *Proc. of ACM SAC*, 2006.

[7] Ch. Bockermann, I. Mierswa, and K. Morik. On the automated creation of understandable positive security models for web applications. In *Proc. of IEEE PerCom*, pages 554–559. IEEE Computer Society, 2008.

[8] O. Shezaf and J. Grossman. Web Hacking Incident Database, 2008.

[9] S. Y. Lee, W. L. Low, and P. Y. Wong. Learning fingerprints for a database intrusion detection system. In *Proc. of ESORICS*, pages 264–280. Springer, 2002.

[10] G. Buehrer, B. W. Weide, and P. A. G. Sivilotti. Using parse tree validation to prevent sql injection attacks. In *Proc. of SEM*, pages 106–113. ACM, 2005.

[11] R. Gerstenberger. Anomaliebasierte Angriffserkennung im FTP-Protokoll. Master's thesis, University of Potsdam, Germany, 2008.

[12] P. Düssel, C. Gehl, P. Laskov, and K. Rieck. Incorporation of application layer protocol syntax into anomaly detection. In *Proc. of Int. Conf. on Information Systems Security (ICISS)*, pages 188–202, 2008.

[13] Y. Hu and B. Panda. A data mining approach for database intrusion detection. In *Proc. of ACM SAC*, pages 711–716. ACM, 2004.

[14] A. Srivastava, S. Sural, and A. K. Majumdar. Database intrusion detection using weighted sequence mining. *JCP*, 1(4):8–17, 2006.

[15] A. Roichman and E. Gudes. DIWeDa - detecting intrusions in web databases. In *Proc. of IFIP Conf. on Data and Appl. Security*, pages 313–329. Springer, 2008.

[16] D. D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *Machine Learning: ECML-98*, pages 4–15. Springer, 1998.

[17] K. Rieck, T. Holz, C. Willems, P. Düssel, and P. Laskov. Learning and classification of malware behaviour. In *Proc. of DIMVA*. Springer, 2008.

[18] D. Haussler. Convolution kernels on discrete structures. Technical report, Dept. of Computer Science, UC Santa Cruz, 1999.

[19] M. Collins and N. Duffy. Convolution kernels for natural language. In *Advances in Neural Information Processing Systems 14*, pages 625–632. MIT Press, 2001.

[20] G. D. Zhou, M. Zhang, D. H. Ji, and Q. M. Zhu. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Proc. of Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 728–736. Assoc. for Computer Linguistics, 2007.

Combining Instance-Based Learning and Logistic Regression for Multilabel Classification (*Resubmission*)[*]

Weiwei Cheng and Eyke Hüllermeier
University of Marburg
{cheng, eyke}@mathematik.uni-marburg.de

Abstract

Multilabel classification is an extension of conventional classification in which a single instance can be associated with multiple labels. Recent research has shown that, just like for standard classification, instance-based learning algorithms relying on the nearest neighbor estimation principle can be used quite successfully in this context. However, since hitherto existing algorithms do not take correlations and interdependencies between labels into account, their potential has not yet been fully exploited. In this paper, we propose a new approach to multilabel classification, which is based on a framework that unifies instance-based learning and logistic regression, comprising both methods as special cases. This approach allows one to capture interdependencies between labels and, moreover, to combine model-based and similarity-based inference for multilabel classification. As will be shown by experimental studies, our approach is able to improve predictive accuracy in terms of several evaluation criteria for multilabel prediction.

1 Introduction

In conventional classification, each instance is assumed to belong to exactly one among a finite set of candidate classes. As opposed to this, the setting of multilabel classification allows an instance to belong to several classes simultaneously or, say, to attach more than one label to a single instance. Problems of this type are ubiquitous in everyday life: At IMDb, a movie can be categorized as *action*, *crime*, and *thriller*; a CNN news report can be tagged as *people* and *political* at the same time; in biology, a typical multilabel learning example is the gene functional prediction problem, where a gene can be associated with multiple functional classes, such as *metabolism*, *transcription*, and *protein synthesis*.

Multilabel classification has received increasing attention in machine learning in recent years, not only due to its practical relevance, but also as it is interesting from a theoretical point of view. In fact, even though it is possible to reduce the problem of multilabel classification to conventional classification in one way or the other and, hence, to apply existing methods for the latter to solve the former, straightforward solutions of this type are usually not optimal. In particular, since the presence or absence of the different class labels has to be predicted *simultaneously*, it is obviously important to exploit correlations and interdependencies between these labels. This is usually not accomplished by simple transformations to standard classification.

Even though quite a number of more sophisticated methods for multilabel classification has been proposed in the literature, the application of *instance-based learning* (IBL) has not been studied very deeply in this context so far. This is a bit surprising, given that IBL algorithms based on the nearest neighbor estimation principle have been applied quite successfully in classification and pattern recognition for a long time [Aha *et al.*, 1991]. A notable exception is the *multilabel k-nearest neighbor* (MLKNN) method that was recently proposed in [Zhang and Zhou, 2007], where it was shown to be competitive to state-of-the-art machine learning methods.

In this paper, we propose a novel approach to multilabel classification, which is based on a framework that unifies instance-based learning and logistic regression, comprising both methods as special cases. This approach overcomes some limitations of existing instance-based multilabel classification methods, including MLKNN. In particular, it allows one to capture interdependencies between the class labels in a proper way.

The rest of this paper is organized as follows: The problem of multilabel classification is introduced in a more formal way in Section 2, and related work is discussed in Section 3. Our novel method is then described in Section 4. Section 5 is devoted to experiments with several benchmark data sets. The paper ends with a summary and some concluding remarks in Section 6.

2 Multilabel Classification

Let $\mathbb{X}$ denote an instance space and let $\mathcal{L} = \{\lambda_1, \lambda_2 \dots \lambda_m\}$ be a finite set of class labels. Moreover, suppose that each instance $\vec{x} \in \mathbb{X}$ can be associated with a subset of labels $L \in 2^{\mathcal{L}}$; this subset is often called the set of *relevant* labels, while the complement $\mathcal{L} \setminus L$ is considered as *irrelevant* for $\vec{x}$. Given training data in the form of a finite set T of observations in the form of tuples $(\vec{x}, L_{\vec{x}}) \in \mathbb{X} \times 2^{\mathcal{L}}$, typically assumed to be drawn independently from an (unknown) probability distribution on $\mathbb{X} \times 2^{\mathcal{L}}$, the goal in multilabel classification is to learn a classifier $h : \mathbb{X} \to 2^{\mathcal{L}}$ that generalizes well beyond these observations in the sense of minimizing the expected prediction loss with respect to a specific loss function; commonly used loss functions will be reviewed in Section 5.3.

[*] With minor changes, this paper has been accepted at The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2009).

Note that multilabel classification can be reduced to a conventional classification problem in a straightforward way, namely by considering each label subset $L \in 2^{\mathcal{L}}$ as a distinct (meta-)class. This approach is referred to as *label powerset* (LP) in the literature. An obvious drawback of this approach is the potentially large number of classes that one has to deal with in the newly generated problem; obviously, this number is $2^{|\mathcal{L}|}$ (or $2^{|\mathcal{L}|} - 1$ if the empty set is excluded as a prediction). This is the reason why LP typically works well if the original label set $\mathcal{L}$ is small but quickly deteriorates for larger label sets. Nevertheless, LP is often used as a benchmark, and we shall also include it in our experiments later on (cf. Section 5).

Another way of reducing multilabel to conventional classification is offered by the *binary relevance* approach. Here, a separate binary classifier h_i is trained for each label $\lambda_i \in \mathcal{L}$, reducing the supervision to information about the presence or absence of this label while ignoring the other ones. For a query instance $\vec{x}$, this classifier is supposed to predict whether λ_i is relevant for $\vec{x}$ ($h_i(\vec{x}) = 1$) or not ($h_i(\vec{x}) = 0$). A multilabel prediction for $\vec{x}$ is then given by $h(\vec{x}) = \{\lambda_i \in \mathcal{L} \,|\, h_i(\vec{x}) = 1\}$. Since binary relevance learning treats every label independently of all other labels, an obvious disadvantage of this approach is that it ignores correlations and interdependencies between labels.

Some of the more sophisticated approaches learn a multilabel classifier h in an indirect way via a scoring function $f : \mathbb{X} \times \mathcal{L} \rightarrow \mathbb{R}$ that assigns a real number to each instance/label combination. The idea is that a score $f(\vec{x}, \lambda)$ is in direct correspondence with the probability that λ is relevant for $\vec{x}$. Given a scoring function of this type, multilabel prediction can be realized via thresholding:

$$h(\vec{x}) = \{\lambda \in \mathcal{L} \,|\, f(\vec{x}, \lambda) \geq t\} ,$$

where $t \in \mathbb{R}$ is a threshold. As a byproduct, a scoring function offers the possibility to produce a ranking of the class labels, simply by ordering them according to their score. Sometimes, this ranking is even more desirable as a prediction, and indeed, there are several evaluation metrics that compare a true label subset with a predicted ranking instead of a predicted label subset (cf. Section 5.3).

3 Related Work

Multilabel classification has received a great deal of attention in machine learning in recent years, and a number of methods has been developed, often motivated by specific types of applications such as text categorization [Schapire and Singer, 2000; Ueda and Saito, 2003; Kazawa *et al.*, 2005; Zhang and Zhou, 2006], computer vision [Boutell *et al.*, 2004], and bioinformatics [Clare and King, 2001; Elisseeff and Weston, 2002; Zhang and Zhou, 2006]. Besides, several well-established methods for conventional classification have been extended to the multi-label case, including support vector machines [Godbole and Sarawagi, 2004; Elisseeff and Weston, 2002; Boutell *et al.*, 2004], neural networks [Zhang and Zhou, 2006], and decision trees [Vens *et al.*, 2008].

In this paper, we are especially interested in instance-based approaches to multilabel classification, i.e., methods based on the nearest neighbor estimation principle [Dasarathy, 1991; Aha *et al.*, 1991]. This interest is largely motivated by the *multilabel k-nearest neighbor* (MLKNN) method that has recently been proposed in [Zhang and Zhou, 2007]. In that paper, the authors show that MLKNN performs quite well in practice. In the concrete experiments presented, MLKNN even outperformed some state-of-the-art model-based approaches to multilabel classification, including RankSVM and AdaBoost.MH [Elisseeff and Weston, 2002; Comite *et al.*, 2003].

MLKNN is a binary relevance learner, i.e., it learns a single classifier h_i for each label $\lambda_i \in \mathcal{L}$. However, instead of using the standard k-nearest neighbor (KNN) classifier as a base learner, it implements the h_i by means of a combination of KNN and Bayesian inference: Given a query instance $\vec{x}$ with unknown multilabel classification $L \subseteq \mathcal{L}$, it finds the k nearest neighbors of $\vec{x}$ in the training data and counts the number of occurrences of λ_i among these neighbors. Considering this number, y, as information in the form of a realization of a random variable Y, the posterior probability of $\lambda_i \in L$ is given by

$$\mathbf{P}(\lambda_i \in L \,|\, Y = y) = \frac{\mathbf{P}(Y = y \,|\, \lambda_i \in L) \cdot \mathbf{P}(\lambda_i \in L)}{\mathbf{P}(Y = y)} , \tag{1}$$

which leads to the decision rule

$$h_i(\vec{x}) = \begin{cases} 1 & \text{if} \quad \begin{array}{r} \mathbf{P}(Y = y \,|\, \lambda_i \in L)\mathbf{P}(\lambda_i \in L) \geq \\ \mathbf{P}(Y = y \,|\, \lambda_i \notin L)\mathbf{P}(\lambda_i \notin L) \end{array} \\ 0 & \text{otherwise} \end{cases}$$

The prior probabilities $\mathbf{P}(\lambda_i \in L)$ and $\mathbf{P}(\lambda_i \notin L)$ as well as the conditional probabilities $\mathbf{P}(Y = y \,|\, \lambda_i \in L)$ and $\mathbf{P}(Y = y \,|\, \lambda_i \notin L)$ are estimated from the training data in terms of corresponding relative frequencies. As an aside, we note that these estimations come with a relatively high computational complexity, since they involve the consideration of all k-neighborhoods of all training instances.

4 Combining IBL and Logistic Regression

In this section, we introduce a machine learning method whose basic idea is to consider the information that derives from examples similar to a query instance as a feature of that instance, thereby blurring the distinction between instance-based and model-based learning to some extent. This idea is put into practice by means of a learning algorithm that realizes instance-based classification as logistic regression.

4.1 KNN Classification

Suppose an instance $\vec{x}$ to be described in terms of features ϕ_i, $i = 1, 2 \ldots n$, where $\phi_i(\vec{x})$ denotes the value of the i-th feature for instance $\vec{x}$. The instance space $\mathbb{X}$ is endowed with a distance measure: $\Delta(\vec{x}, \vec{x}')$ is the distance between instances $\vec{x}$ and $\vec{x}'$. We shall first focus on the case of binary classification and hence define the set of class labels by $\mathcal{Y} = \{-1, +1\}$. A tuple $(\vec{x}, y) \in \mathbb{X} \times \mathcal{Y}$ is called a labeled instance or example. $\mathcal{D}$ denotes a sample that consists of N labeled instances $(\vec{x}_i, y_i)$, $1 \leq i \leq N$. Finally, a new instance $\vec{x}_0 \in \mathbb{X}$ (a query) is given, whose label $y_0 \in \{-1, +1\}$ is to be estimated.

The nearest neighbor (NN) principle prescribes to estimate the label of the yet unclassified query $\vec{x}_0$ by the label of the nearest (least distant) sample instance. The KNN approach is a slight generalization, which takes the $k \geq 1$ nearest neighbors of $\vec{x}_0$ into account. That is, an estimation $\hat{y}_0$ of y_0 is derived from the set $\mathcal{N}_k(\vec{x}_0)$ of the k nearest neighbors of $\vec{x}_0$, usually by means of a *majority vote*:

$$\hat{y}_0 = \arg\max_{y \in \mathcal{Y}} \#\{\vec{x}_i \in \mathcal{N}_k(\vec{x}_0) \,|\, y_i = y\}. \tag{2}$$

4.2 IBL as Logistic Regression

A key idea of our approach is to consider the labels of neighbored instances as "features" of the query $\vec{x}_0$ whose label is to be estimated. It is worth mentioning that similar ideas have recently been exploited in relational learning [Getoor and Taskar, 2007] and collective classification [Lu and Getoor, 2003; Ghamrawi and McCallum, 2005].

Denote by p_0 the prior probability of $y_0 = +1$ and by π_0 the corresponding posterior probability. Moreover, let $\delta_i \overset{\text{df}}{=} \Delta(\vec{x}_0, \vec{x}_i)$ be the distance between $\vec{x}_0$ and $\vec{x}_i$. Taking the known label y_i as information about the unknown label y_0, we can consider the posterior probability

$$\pi_0 \overset{\text{df}}{=} \mathbf{P}(y_0 = +1 \,|\, y_i).$$

More specifically, Bayes' rule yields

$$\begin{aligned}\frac{\pi_0}{1-\pi_0} &= \frac{\mathbf{P}(y_i \,|\, y_0 = +1)}{\mathbf{P}(y_i \,|\, y_0 = -1)} \cdot \frac{p_0}{1-p_0} \\ &= \rho \cdot \frac{p_0}{1-p_0},\end{aligned}$$

where ρ is the likelihood ratio. Taking logarithms on both sides, we get

$$\log\left(\frac{\pi_0}{1-\pi_0}\right) = \log(\rho) + \omega_0 \tag{3}$$

with $\omega_0 = \log(p_0) - \log(1-p_0)$.

Model (3) still requires the specification of the likelihood ratio ρ. In order to obey the basic principle underlying IBL, the latter should be a function of the distance δ_i. In fact, ρ should become large for $\delta_i \to 0$ if $y_i = +1$ and small if $y_i = -1$: Observing a very close instance $\vec{x}_i$ with label $y_i = +1$ ($y_i = -1$) makes $y_0 = +1$ more (un)likely in comparison to $y_i = -1$. Moreover, ρ should tend to 1 as $\delta_i \to \infty$: If $\vec{x}_i$ is too far away, its label does not provide any evidence, neither in favor of $y_0 = +1$ nor in favor of $y_0 = -1$. A parameterized function satisfying these properties is

$$\rho = \rho(\delta) \overset{\text{df}}{=} \exp\left(y_i \cdot \frac{\alpha}{\delta}\right),$$

where $\alpha > 0$ is a constant. Note that the choice of a special functional form for ρ is quite comparable to the specification of the kernel function used in (non-parametric) kernel-based density estimation, as well as to the choice of the weight function in weighted NN estimation. $\rho(\delta)$ actually determines the probability that two instances whose distance is given by $\delta = \Delta(\vec{x}_0, \vec{x}_i)$ do have the same label.

Now, taking the complete sample neighborhood $\mathcal{N}(\vec{x}_0)$ of $\vec{x}_0$ into account and —as in the naive Bayes approach— making the simplifying assumption of conditional independence, we obtain

$$\begin{aligned}\log\left(\frac{\pi_0}{1-\pi_0}\right) &= \omega_0 + \alpha \sum_{\vec{x}_i \in \mathcal{N}(\vec{x}_0)} \frac{y_i}{\delta_i} \\ &= \omega_0 + \alpha \cdot \omega_+(\vec{x}_0),\end{aligned} \tag{4}$$

where $\omega_+(\vec{x}_0)$ can be seen as a summary of the evidence in favor of label $+1$. As can be seen, the latter is simply given by the sum of neighbors with label $+1$, weighted by their distance, minus the weighted sum of neighbors with label -1.

As concerns the classification of the query $\vec{x}_0$, the decision is determined by the sign of the right-hand side in (4). From this point of view, (4) does basically realize a weighted NN estimation, or, stated differently, it is a "model-based" version of instance-based learning. Still, it differs from the simple NN scheme in that it includes a bias term ω_0, which plays the same role as the prior probability in Bayesian inference.

From a statistical point of view, (4) is nothing else than a logistic regression equation. In other words, taking a "feature-based" view of instance-based learning and applying a Bayesian approach to inference comes down to realizing IBL as logistic regression.

By introducing a *similarity measure* κ, inversely related to the distance function Δ, (4) can be written in the form

$$\log\left(\frac{\pi_0}{1-\pi_0}\right) = \omega_0 + \alpha \sum_{\vec{x}_i \in \mathcal{N}(\vec{x}_0)} \kappa(\vec{x}_0, \vec{x}_i) \cdot y_i \ . \tag{5}$$

Note that, as a special case, this approach can mimic the standard KNN classifier (2), namely by setting $\omega_0 = 0$ and defining κ in terms of the (data-dependent) "KNN kernel"

$$\kappa(\vec{x}_0, \vec{x}_i) = \begin{cases} 1 & \text{if } \vec{x}_i \in \mathcal{N}_k(\vec{x}_0) \\ 0 & \text{otherwise} \end{cases} . \tag{6}$$

4.3 Estimation and Classification

The parameter α in (4) determines the weight of the evidence

$$\omega_+(\vec{x}_0) = \sum_{\vec{x}_i \in \mathcal{N}(\vec{x}_0)} \kappa(\vec{x}_0, \vec{x}_i) \cdot y_i \tag{7}$$

and, hence, its influence on the posterior probability estimation π_0. In fact, α plays the role of a smoothing (regularization) parameter. The smaller α is chosen, the smoother an estimated probability function (obtained by applying (5) to all points $\vec{x}_0 \in \mathcal{X}$) will be. In the extreme case where $\alpha = 0$, one obtains a constant function (equal to ω_0).

An optimal specification of α can be accomplished by adapting this parameter to the data $\mathcal{D}$, using the method of maximum likelihood (ML). For each sample point $\vec{x}_j$ denote by

$$\omega_+(\vec{x}_j) \overset{\text{df}}{=} \sum_{\vec{x}_j \neq \vec{x}_i \in \mathcal{N}(\vec{x}_j)} \kappa(\vec{x}_i, \vec{x}_j) \cdot y_i$$

the sample evidence in favor of $y_j = +1$. The log-likelihood function is then given by the mapping

$$\alpha \mapsto \sum_{j \,:\, y_j = +1} w_0 + \alpha\, \omega_+(\vec{x}_j) - \sum_{j=1}^{N} \log\left(1 + \exp(w_0 + \alpha\, \omega_+(\vec{x}_j)\right), \tag{8}$$

and the optimal parameter α^* is the maximizer of (8). The latter can be computed by means of standard methods from logistic regression. The posterior probability π_0 for the query is then given by

$$\pi_0 = \frac{\exp(\omega_0 + \alpha^*\, \omega_+(\vec{x}_0))}{1 + \exp(\omega_0 + \alpha^*\, \omega_+(\vec{x}_0))} \ .$$

To classify $\vec{x}_0$, one applies the decision rule

$$\hat{y}_0 \overset{\text{df}}{=} \begin{cases} +1 & \text{if} \quad \pi_0 \geq 1/2 \\ -1 & \text{if} \quad \pi_0 < 1/2 \end{cases} .$$

Subsequently, we shall refer to the method outlined above as IBLR (Instance-Based Learning by Logistic Regression).

4.4 Including Additional Features

In the previous section, instance-based learning has been embedded into logistic regression, using the information coming from the neighbors of a query $\vec{x}_0$ as a "feature" of that query. In this section, we consider a possible generalization of this approach, namely the idea to extend the model (5) by taking further features of $\vec{x}_0$ into account:

$$\log\left(\frac{\pi_0}{1-\pi_0}\right) = \alpha\,\omega_+(\vec{x}_0) + \sum_{\varphi_s\in\mathcal{F}} \beta_s\,\varphi_s(\vec{x}_0), \quad (9)$$

where $\mathcal{F} = \{\varphi_0, \varphi_1 \dots \varphi_r\}$ is a subset of the available features $\{\phi_0, \phi_1 \dots \phi_n\}$ and $\varphi_0 = \phi_0 \equiv 1$, which means that β_0 plays the role of ω_0. Equation (9) is a common logistic regression model, except that $\omega_+(\vec{x}_0)$ is a "non-standard" feature.

The approach (9), that we shall call IBLR+, integrates instance-based and model-based (attribute-based) learning and, by estimating the regression coefficients in (9), achieves an optimal balance between both approaches. The extended model (9) can be interpreted as a logistic regression model of IBL, as outlined in Section 4.2, where the bias ω_0 is no longer constant:

$$\log\left(\frac{\pi_0}{1-\pi_0}\right) = \omega_0(\vec{x}_0) + \alpha\,\omega_+(\vec{x}_0)\ , \quad (10)$$

with $\omega_0(\vec{x}_0) \overset{\text{df}}{=} \sum \beta_s\varphi_s(\vec{x}_0)$ being an instance-specific bias determined by the model-based part of (9).

4.5 Extension to Multilabel Classification

So far, we only considered the case of binary classification. To extend the approach to multilabel classification with a label set $\mathcal{L} = \{\lambda_1, \lambda_2 \dots \lambda_m\}$, the idea is to train one classifier h_i for each label. For the i-th label λ_i, this classifier is derived from the model

$$\log\left(\frac{\pi_0^{(i)}}{1-\pi_0^{(i)}}\right) = \omega_0^{(i)} + \sum_{j=1}^{m} \alpha_j^{(i)} \cdot \omega_{+j}^{(i)}(\vec{x}_0)\ , \quad (11)$$

where $\pi_0^{(i)}$ denotes the (posterior) probability that λ_i is relevant for $\vec{x}_0$, and

$$\omega_{+j}^{(i)}(\vec{x}_0) = \sum_{\vec{x}\in\mathcal{N}(\vec{x}_0)} \kappa(\vec{x}_0, \vec{x}) \cdot y_j(\vec{x}) \quad (12)$$

is a summary of the presence of the j-th label λ_j in the neighborhood of $\vec{x}_0$; here, $y_j(\vec{x}) = +1$ if λ_j is present (relevant) for the neighbor $\vec{x}$, and $y_j(\vec{x}) = -1$ in case it is absent (non-relevant).

Obviously, the approach (11) is able to take interdependencies between class labels into consideration. More specifically, the estimated coefficient $\alpha_j^{(i)}$ indicates to what extent the relevance of label λ_i is influenced by the relevance of λ_j. A value $\alpha_j^{(i)} \gg 0$ means that the presence of λ_j makes the relevance of λ_i more likely, i.e., there is a positive correlation. Correspondingly, a negative coefficient would indicate a negative correlation.

Note that the estimated probabilities $\pi_0^{(i)}$ can naturally be considered as scores for the labels λ_i. Therefore, a ranking of the labels is simply obtained by sorting them in decreasing order according to their probabilities. Moreover, a pure multilabel prediction for $\vec{x}_0$ is derived from this ranking via thresholding at $t = 0.5$.

Of course, it is also possible to combine the model (11) with the extension proposed in Section 4.4. This leads to a model

$$\log\left(\frac{\pi_0^{(i)}}{1-\pi_0^{(i)}}\right) = \sum_{j=1}^{m} \alpha_j^{(i)} \cdot \omega_{+j}^{(i)}(\vec{x}_0) + \sum_{\varphi_s\in\mathcal{F}} \beta_s^{(i)}\,\varphi_r(\vec{x}_0)\ . \quad (13)$$

We shall refer to the extensions (11) and (13) of IBLR to multilabel classification as IBLR-ML and IBLR-ML+, respectively.

5 Experimental Results

This section is devoted to experimental studies that we conducted to get a concrete idea of the performance of our method. Before presenting the results of our experiments, we give some information about the learning algorithms and data sets included in the study, as well as the criteria used for evaluation.

5.1 Learning Algorithms

For the reasons mentioned previously, our main interest is focused on MLKNN, which is arguably the state-of-the-art in instance-based multilabel ranking. This method is parameterized by the size of the neighborhood, for which we adopted the value $k = 10$. This value is recommended in [Zhang and Zhou, 2007], where it was found to yield the best performance. For the sake of fairness, we use the same neighborhood size for our method, in conjunction with the KNN kernel (6). In both cases, the simple Euclidean metric (on the complete attribute space) was used as a distance function. For our method, we tried both variants, the pure instance-based version (11), and the extended model (13) with $\mathcal{F}$ including all available features. Intuitively, one may expect the latter, IBLR-ML+, to be advantageous to the former, IBLR-ML, as it can use features in a more flexible way. Yet, one should note that, since we simply included all attributes in $\mathcal{F}$, each attribute will essentially be used twice in IBLR-ML+, thus producing a kind of redundancy. Besides, model induction will of course become more difficult, since a larger number of parameters needs to be estimated.

As an additional baseline we used binary relevance learning (BR) with three different base learners: logistic regression, C4.5 (the WEKA [Witten and Frank, 2005] implementation J48 in its default setting), and KNN (again with $k = 10$). Finally, we also included label powerset (LP) with C4.5 as a base learner.

5.2 Data Sets

Benchmark data for multi-label classification is not as abundant as for conventional classification, and indeed, experiments in this field are often restricted to a very few or even only a single data set. For our experimental study, we have collected a comparatively large number of seven data sets from different domains; an overview is given in Table 1.[1]

5.3 Evaluation Measures

To evaluate the performance of multilabel classification methods, a number of criteria and metrics have been proposed in the literature. For a classifier h, let $h(\vec{x}) \subseteq \mathcal{L}$ denote its multilabel prediction for an instance $\vec{x}$, and let $L_{\vec{x}}$

[1] All data sets are public available at `http://mlkd.csd.auth.gr/multilabel.html` and `http://lamda.nju.edu.cn/data.htm`.

Table 1: Statistics for the multilabel data sets used in the experiments. The symbol * indicates that the data set contains binary features; *card* (cardinality) is the average number of labels per instance.

DATA SET	DOMAIN	#INST	#ATTR	#LABEL	CARD
emotions	music	593	72	6	1.87
image	vision	2000	135	5	1.24
genbase	biology	662	1186*	27	1.25
mediamill	multimedia	5000	120	101	4.27
reuters	text	7119	243	7	1.24
scene	vision	2407	294	6	1.07
yeast	biology	2417	103	14	4.24

denote the true set of relevant labels. Moreover, in case a related scoring function f is also defined, let $f(\vec{x}, \lambda)$ denote the score assigned to label λ for instance $\vec{x}$. The most commonly used evaluation measures are defined as follows:

- *Hamming loss* computes the percentage of labels whose relevance is predicted incorrectly:

$$\textsc{HamLoss}(h) = \frac{1}{|\mathcal{L}|} \big| h(\vec{x}) \, \Delta \, L_{\vec{x}} \big|, \tag{14}$$

  where Δ is the symmetric difference between two sets.

- *One error* computes how many times the top-ranked label is not relevant:

$$\textsc{OneError}(f) = \begin{cases} 1 & \text{if } \arg\max_{\lambda \in \mathcal{L}} f(\vec{x}, \lambda) \notin L_{\vec{x}} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

- *Coverage* determines how far one needs to go in the list of labels to cover all the relevant labels of an instance. This measure is loosely related to the precision at the level of perfect recall:

$$\textsc{Coverage}(f) = \max_{\lambda \in L_{\vec{x}}} rank_f(\vec{x}, \lambda) - 1 \ , \tag{16}$$

  where $rank_f(\vec{x}, \lambda)$ denotes the position of label $\vec{x}$ in the ordering induced by f.

- *Rank loss* computes the average fraction of label pairs that are not correctly ordered:

$$\textsc{RankLoss}(f) = \tag{17}$$

$$\frac{\#\{(\lambda, \lambda') \,|\, f(\vec{x}, \lambda) \leq f(\vec{x}, \lambda'), (\lambda, \lambda') \in L_{\vec{x}} \times \overline{L_{\vec{x}}}\}}{|L_{\vec{x}}||\overline{L_{\vec{x}}}|}, \tag{18}$$

  where $\overline{L_{\vec{x}}} = \mathcal{L} \setminus L_{\vec{x}}$ is the set of irrelevant labels.

- *Average precision* determines for each relevant label $\lambda \in L_{\vec{x}}$ the percentage of relevant labels among all labels that are ranked above it, and averages these percentages over all relevant labels:

$$\textsc{AvePrec}(f) = \tag{19}$$

$$\frac{1}{|L_{\vec{x}}|} \sum_{\lambda \in L_{\vec{x}}} \frac{|\{\lambda' \,|\, rank_f(\vec{x}, \lambda') \leq rank_f(\vec{x}, \lambda), \lambda' \in L_{\vec{x}}\}|}{rank_f(\vec{x}, \lambda)}. \tag{20}$$

Notice that only Hamming loss evaluates mere multilabel predictions (i.e., the multilabel classifier h), while the others metrics evaluate the underlying ranking function f. Moreover, smaller values indicate better performance for all measures except average precision. Finally, except for coverage, all measures are normalized and assume values between 0 and 1.

5.4 Results and Discussion

The results of a cross validation study (10-fold, 5 repeats) are summarized in Table 3 at the end of the paper. As can be seen, the baseline methods BR and LP are in general not competitive. Looking at the average ranks, IBLR-ML consistently outperforms all other methods, regardless of the evaluation metric, indicating that it is the strongest method overall. The ranking among the three instance-based methods is IBLR-ML $\succ$ IBLR-ML+ $\succ$ MLKNN for all measures except ONEERROR, for which the latter two change the position.

To analyze the results more thoroughly, we followed the two-step statistical test procedure recommended in [Demsar, 2006], consisting of a Friedman test of the null hypothesis that all learners have equal performance and, in case this hypothesis is rejected, a Nemenyi test to compare learners in a pairwise way. Both tests are based on the average ranks as shown in the bottom line in Table 3. Even though the Friedman test suggests that there are significant differences between the methods, most of the pairwise comparisons remain statistically non-significant (at a significance level of 5%); see Fig. 1. This is not surprising, however, given that the number of data sets included in the experiments, despite being much higher than usual, is still quite limited from a statistical point of view. Nevertheless, the overall picture taken from the experiments is clearly in favor of IBLR-ML.

Table 2: Classification error on binary classification problems. The number in brackets behind the performance value is the rank of the method on the corresponding data set (for each data set, the methods are ranked in decreasing order of performance). The average rank is the average of the ranks across all data sets.

DATA SET	IBLR-ML+	IBLR-ML	MLKNN	BR-KNN
breast-cancer	.280(4)	.252(1)	.259(2)	.262(3)
breast-w	.037(3.5)	.037(3.5)	.036(2)	.034(1)
colic	.195(3)	.176(1)	.350(4)	.182(2)
credit-a	.135(2)	.132(1)	.328(4)	.138(3)
credit-g	.229(1)	.265(3)	.306(4)	.261(2)
diabetes	.233(1)	.263(4)	.259(3)	.256(2)
heart-statlog	.170(1)	.193(2.5)	.363(4)	.193(2.5)
hepatitis	.175(1)	.192(2)	.204(4)	.199(3)
ionosphere	.117(2.5)	.117(2.5)	.108(1)	.171(4)
kr-vs-kp	.018(1)	.044(2.5)	.044(2.5)	.046(4)
labor	.210(3)	.130(1)	.270(4)	.150(2)
mushroom	.000(1.5)	.000(1.5)	.001(3.5)	.001(3.5)
sick	.030(1)	.039(2)	.061(4)	.040(3)
sonar	.250(2)	.245(1)	.327(4)	.284(3)
tic-tac-toe	.125(1)	.137(3)	.136(2)	.317(4)
vote	.044(1)	.060(2)	.074(3)	.076(4)
average rank	1.84	2.09	3.19	2.88

As to MLKNN, it is interesting to compare this method with the BR-version of KNN. In fact, since MLKNN is a binary relevance learner, too, the only difference between these two methods concerns the incorporation of global information in MLKNN, which is accomplished through the Bayesian updating (1) of local information about the relevance of labels. From Table 3, it can be seen that MLKNN is better than BR-KNN in terms of all ranking measures, but not in terms of the Hamming loss, for which it is even a bit worse. Thus, in terms of mere relevance prediction, MLKNN does not seem to offer special advantages. Our explanation for this finding is that the incorporation of

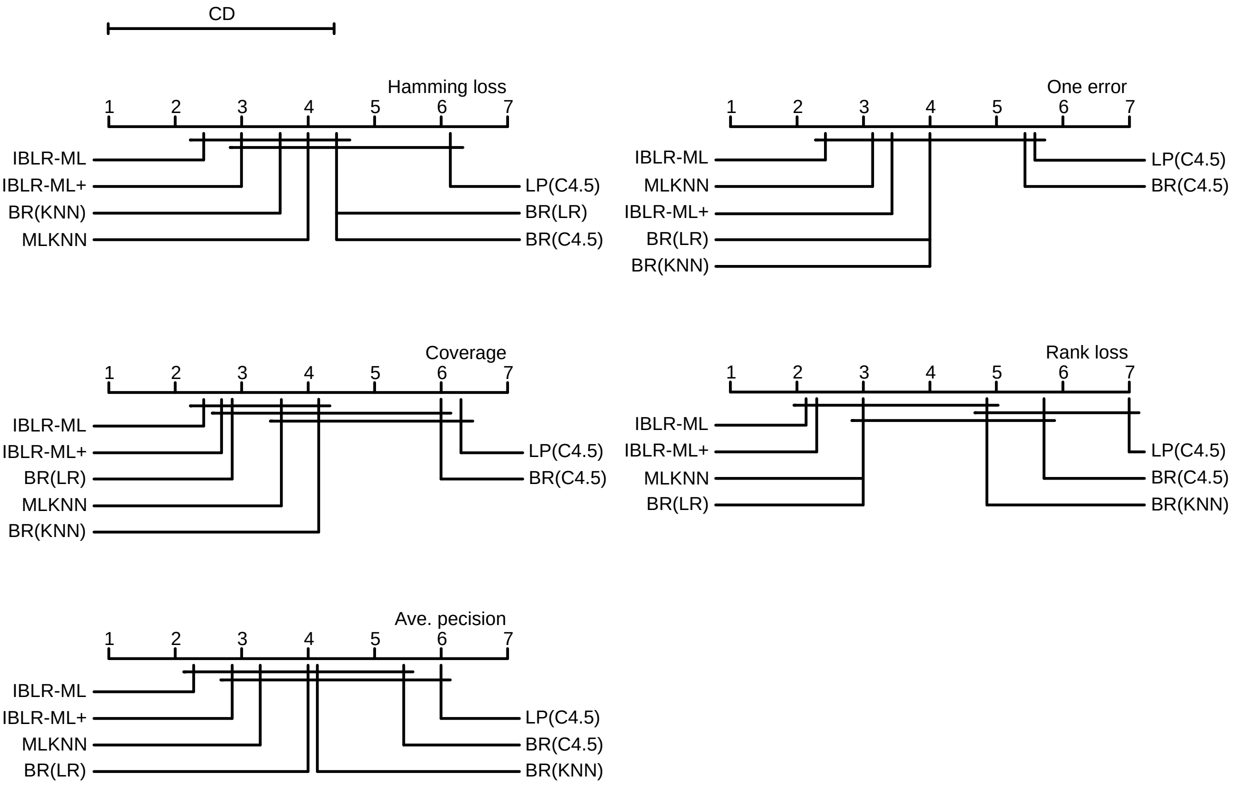


Figure 1: Comparison of all classifiers against each other with the Nemenyi test. Groups of classifiers that are not significantly different (at $p = 0.05$) are connected.

global information is indeed not useful for a simple 0/1 prediction. In a sense, this is perhaps not very surprising, given that the use of global information is somehow in conflict with the basic principle of local estimation underlying nearest neighbor prediction. Exploiting such information does, however, offer a reasonable way *to break ties between class labels*, which in turn explains the positive effect on ranking performance. In fact, one should note that, when simply scoring labels by the number of occurrences among the k neighbors of a query, such ties are quite likely; in particular, all non-relevant labels that never occur will have a score of 0 and will hence be tied. Resorting to global information about their relevance is then clearly more reasonable than breaking ties at random.

To validate our conjecture that the incorporation of global information in MLKNN is actually not very useful for mere relevance prediction, we have conducted an additional experiments using 16 binary classification problems from the UCI repository. Using this type of data makes sense, since, for a binary relevance learner, minimizing Hamming loss is equivalent to minimizing 0/1 loss for m binary classification problems that are solved independently of each other. The results of a 5 times 10-fold cross validation, summarized in Table 2, are completely in agreement with our previous study. MLKNN does indeed show the worst performance and is even outperformed by the simple BR-KNN. Interestingly, IBLR-ML+ is now a bit better than IBLR-ML. A reasonable explanation for this finding is that, compared to the multi-label case, the relevance information that comes from the neighbors of a query in binary classification only concerns a single label and, therefore, is rather sparse. Correspondingly, information about additional features is revaluated.

6 Summary and Conclusions

We have presented a novel approach to instance-based learning, called IBLR, that can be used for classification in general and for multilabel classification in particular. Considering label information of neighbored examples as features of a query instance, the idea of IBLR is to reduce instance-based learning formally to logistic regression. An optimal balance between global and local inference, and in the extended version IBLR+ also between instance-based and model-based (attribute-oriented) learning, can then be achieved by the estimation of optimal regression coefficients.

For multilabel classification, this idea is especially appealing, as it allows one to take interdependencies between different labels into consideration. These dependencies are directly reflected by the sign and magnitude of related regression coefficients. This ability distinguishes IBLR from hitherto existing instance-based methods for multilabel classification, and is probably one of the main factors for its excellent performance. In fact, our extensive empirical study has clearly shown that IBLR improves upon existing methods, in particular the MLKNN method that can be considered as the state-of-the-art in instance-based multilabel classification.

Interestingly, our results also suggest that the basic idea underlying MLKNN, namely to combine instance-based learning and Bayesian inference, is beneficial for the ranking performance but not in terms of mere relevance prediction. Investigating the influence on specific performance measures in more detail, and elaborating on (instance-based) methods for minimizing specific loss functions, is an interesting topic of future work. Besides, for IBLR+, we plan to exploit the possibility to combine instance-based and model-based inference in a more sophisticated way, for example by selecting optimal feature subsets for both parts instead of simply using all features twice.

References

[Aha *et al.*, 1991] D. Aha, D. Kibler, and M. Alber. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.

[Boutell *et al.*, 2004] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classiffication. *Pattern Recognition*, 37(9):1757–1771, 2004.

[Clare and King, 2001] A. Clare and R. D. King. Knowledge discovery in multi-label phenotype data. In L. D. Raedt and A. Siebes, editors, *Lecture Notes in Computer Science*, volume 2168, pages 42–53, Berlin, 2001. Springer.

[Comite *et al.*, 2003] F. D. Comite, R. Gilleron, and M. Tommasi. Learning multi-label alternating decision tree from texts and data. In P. Perner and A. Rosenfeld, editors, *Lecture Notes in Computer Science*, volume 2734, pages 35–49, Berlin, 2003. Springer.

[Dasarathy, 1991] B.V. Dasarathy, editor. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, California, 1991.

[Demsar, 2006] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[Elisseeff and Weston, 2002] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14, pages 681–687, Cambridge MA, 2002. MIT Press.

[Getoor and Taskar, 2007] Lise Getoor and Ben Taskar, editors. *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[Ghamrawi and McCallum, 2005] Nadia Ghamrawi and Andrew McCallum. Collective multi-label classification. In *Proc. CIKM-05*, Bremen, Germany, 2005.

[Godbole and Sarawagi, 2004] S. Godbole and S. Sarawagi. Discriminative methods for multi-labeled classiffication. In *Advances in Knowledge Discovery and Data Mining*, volume 3056 of *LNCS*, pages 20–33. Springer, 2004.

[Kazawa *et al.*, 2005] H. Kazawa, T. Izumitani, H. Taira, and E. Maeda. Maximal margin labeling for multi-topic text categorization. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Inf. Proc. Syst.*, volume 17, Cambridge MA, 2005. MIT Press.

[Lu and Getoor, 2003] Q. Lu and L. Getoor. Link-based classification. In *Proc. ICML-03*, pages 496–503, Washington, 2003.

[Schapire and Singer, 2000] R. E. Schapire and Y. Singer. Boostexter: a boosting-based system for text categorization. *Machine Learning*, 39(2):135–168, 2000.

[Ueda and Saito, 2003] N. Ueda and K. Saito. Parametric mixture models for multi-label text. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing*, volume 15, pages 721–728, Cambridge MA, 2003. MIT Press.

[Vens *et al.*, 2008] C. Vens, J. Struyf, L. Schietgat, S. Dzeroski, and H. Blockeel. Decision trees for hierarchical multi-label classification. *Machine Learning*, 73:185–214, 2008.

[Witten and Frank, 2005] Ian Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, CA, USA, 2nd edition, 2005.

[Zhang and Zhou, 2006] M.-L. Zhang and Z.-H. Zhou. Multi-label neural networks with applications to functional genomics and text categorization. In *IEEE Transactions on Knowledge and Data Engineering*, volume 18, pages 1338–1351, 2006.

[Zhang and Zhou, 2007] M.-L. Zhang and Z.-H. Zhou. ML-kNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

Table 3: Experimental results in terms of different evaluation measures. The number in brackets behind the performance value is the rank of the method on the corresponding data set (for each data set, the methods are ranked in decreasing order of performance). The average rank is the average of the ranks across all data sets.

	IBLR-ML+	IBLR-ML	MLKNN	LP	BR-LR	BR-C4.5	BR-KNN
Hamming							
emotions	0.213(3)	0.185(1)	0.263(6)	0.265(7)	0.214(4)	0.253(5)	0.191(2)
genbase	0.002(2)	0.002(3)	0.005(7)	0.002(4)	0.002(5)	0.001(1)	0.004(6)
image	0.182(1)	0.189(2)	0.195(4)	0.257(7)	0.202(5)	0.245(6)	0.193(3)
mediamill	0.03(6)	0.028(3)	0.027(2)	0.039(7)	0.029(4)	0.032(5)	0.027(1)
reuters	0.044(1)	0.084(6)	0.073(5)	0.067(4)	0.049(2)	0.058(3)	0.09(7)
scene	0.126(4)	0.084(1)	0.087(2)	0.142(7)	0.14(6)	0.133(5)	0.093(3)
yeast	0.199(4)	0.194(1)	0.194(2)	0.28(7)	0.206(5)	0.25(6)	0.196(3)
average rank	3	2.43	4	6.14	4.43	4.43	3.57
One Error							
emotions	0.278(3)	0.257(1)	0.393(5)	0.43(7)	0.278(4)	0.422(6)	0.265(2)
genbase	0.014(5)	0.007(2)	0.009(3)	0.01(4)	0.015(6)	0.003(1)	0.017(7)
image	0.328(1)	0.367(2)	0.382(4)	0.507(6)	0.37(3)	0.512(7)	0.386(5)
mediamill	0.356(5)	0.185(3)	0.136(2)	0.367(6)	0.277(4)	0.381(7)	0.133(1)
reuters	0.076(1)	0.22(6)	0.185(5)	0.162(4)	0.086(2)	0.145(3)	0.233(7)
scene	0.349(4)	0.224(2)	0.223(1)	0.394(6)	0.364(5)	0.411(7)	0.26(3)
yeast	0.249(5)	0.227(1)	0.228(2)	0.351(6)	0.241(4)	0.389(7)	0.234(3)
average rank	3.43	2.43	3.14	5.57	4	5.43	4
Coverage							
emotions	1.844(4)	1.689(1)	2.258(5)	2.576(6)	1.836(3)	2.608(7)	1.771(2)
genbase	0.356(1)	0.422(4)	0.561(7)	0.529(6)	0.391(3)	0.372(2)	0.436(5)
image	0.963(1)	1.056(3)	1.129(5)	1.589(6)	1.052(2)	1.615(7)	1.102(4)
mediamill	16.681(4)	15.161(3)	12.757(1)	49.469(7)	14.323(2)	47.996(6)	21.344(5)
reuters	0.411(1)	0.758(4)	0.676(3)	0.986(7)	0.44(2)	0.852(6)	0.82(5)
scene	0.911(5)	0.466(1)	0.472(2)	1.145(6)	0.871(4)	1.288(7)	0.551(3)
yeast	6.289(3)	6.203(1)	6.273(2)	9.204(6)	6.492(4)	9.353(7)	6.517(5)
average rank	2.71	2.43	3.57	6.29	2.86	6	4.14
Rank Loss							
emotions	0.168(2)	0.146(1)	0.258(5)	0.499(7)	0.168(3)	0.372(6)	0.183(4)
genbase	0.002(1)	0.004(2)	0.006(4)	0.017(7)	0.005(3)	0.006(5)	0.01(6)
image	0.175(1)	0.197(3)	0.214(4)	0.537(7)	0.196(2)	0.409(6)	0.252(5)
mediamill	0.05(4)	0.043(3)	0.037(1)	0.451(7)	0.041(2)	0.187(6)	0.117(5)
reuters	0.026(1)	0.083(4)	0.069(3)	0.18(7)	0.03(2)	0.092(5)	0.113(6)
scene	0.15(4)	0.076(1)	0.077(2)	0.393(7)	0.157(5)	0.299(6)	0.109(3)
yeast	0.168(3)	0.164(1)	0.167(2)	0.545(7)	0.176(4)	0.362(6)	0.204(5)
average rank	2.29	2.14	3	7	3	5.71	4.86
Ave. Prec.							
emotions	0.794(3)	0.816(1)	0.71(5)	0.683(6)	0.794(4)	0.683(7)	0.805(2)
genbase	0.989(3)	0.99(2)	0.989(4)	0.986(6)	0.988(5)	0.993(1)	0.982(7)
image	0.789(1)	0.763(2)	0.748(5)	0.653(6)	0.763(3)	0.649(7)	0.752(4)
mediamill	0.694(5)	0.731(3)	0.751(1)	0.498(7)	0.722(4)	0.582(6)	0.739(2)
reuters	0.951(1)	0.859(6)	0.881(4)	0.871(5)	0.944(2)	0.889(3)	0.848(7)
scene	0.773(4)	0.867(1)	0.867(2)	0.734(6)	0.769(5)	0.715(7)	0.844(3)
yeast	0.763(3)	0.769(1)	0.764(2)	0.621(6)	0.754(5)	0.619(7)	0.761(4)
average rank	2.86	2.29	3.29	6	4	5.43	4.14

Extension and Empirical Comparison of Graph-Kernels for the Analysis of Protein Active Sites

Thomas Fober[*], Marco Mernberger[*], Vitalik Melnikov, Ralph Moritz, Eyke Hüllermeier
Department of Mathematics and Computer Science
Marburg University, Germany
{thomas,mernberger,melnikov,moritz,eyke}@mathematik.uni-marburg.de

Abstract

Graphs are often used to describe and analyze the geometry and physicochemical composition of biomolecular structures, such as chemical compounds and protein active sites. A key problem in graph-based structure analysis is to define a measure of similarity that enables a meaningful comparison of such structures. In this regard, so-called kernel functions have recently attracted a lot of attention, especially since they allow for the application of a rich repertoire of methods from the field of kernel-based machine learning. Most of the existing kernel functions on graph structures, however, have been designed for the case of unlabeled and/or unweighted graphs. Since proteins are often more naturally and more exactly represented in terms of node-labeled and edge-weighted graphs, we propose corresponding extensions of existing graph kernels. Moreover, we propose an instance of the substructure fingerprint kernel suitable for the analysis of protein binding sites. The performance of these kernels is investigated by means of an experimental study in which graph kernels are used as similarity measures in the context of classification.

1 Introduction

The functional analysis of proteins is a key research problem in the life sciences and a main prerequisite for resolving the proteome and interactome of living cells, tissues and organisms. Since improved technology has led to an increased number of known protein structures, structure-based prediction of protein function has now become a viable alternative to classical sequence-based prediction methods. In fact, structure-based approaches complement sequence-based methods in a reasonable way, as it is well-known that functional similarity does not necessarily come along with sequence similarity [Gibrat *et al.*, 1996].

Prediction of protein function can be seen as a classification problem. In machine learning, a large repertoire of classification methods has been developed, most of them relying, in one way or the other, on a kind of similarity measure between the objects to be classified. What is needed, therefore, is a measure of similarity between protein structures. More specifically, our focus in this paper will be on the special case of *protein binding sites* derived from crystal structures. To model such structures in a formal way, we resort to a graph representation which is able to capture the most important geometrical and physicochemical properties of a binding site.

For a long time, graphs have been used in chemoinformatics for the modeling of chemical compounds [Bunke and Jiang, 2000]. In bioinformatics, they are becoming more and more important, too, due to their general versatility in modeling complex structures such as proteins or interaction networks [Berg and Lässig, 2004]. It is hence not surprising that a number of methods has been developed for comparing graphs representing protein structures (e.g. [Jambon *et al.*, 2003; Weskamp *et al.*, 2007; Fober *et al.*, 2009]), and for computing related similarity measures, for example based the concepts of maximum (minimum) common subgraph (supergraph) [Raymond *et al.*, 2002; Raymond and Willett, 2002] or graph edit distance [Neuhaus and Bunke, 2007].

In this context, so-called *kernel functions* (on graphs) have attracted increasing attention in recent years [Gärtner, 2003]. Here, the term ‘kernel’ refers to a class of functions that fulfill certain mathematical properties and can typically be interpreted as similarity measures. These functions are especially attractive as they can be used as a ‘plug-in’ for every kernel-based machine learning method. In other words, as soon as a kernel function has been defined on a certain class of objects, the related domain becomes amenable to these methods.

The random walk kernel [Gärtner, 2003] and the shortest path kernel [Borgwardt, 2007] are among the most prominent graph kernels that have been used in the fields of bio- or chemoinformatics. However, as they have originally been defined for unweighted graphs, they are not immediately applicable to the case of graphs modeling protein binding sites. In fact, as will be explained in more detail in Section 2, binding sites are more naturally modeled in terms of graphs with node labels and edge weights, and a representation ignoring labels and weights would come along with an unacceptable loss of information. In Section 3, we therefore extend the aforementioned kernel functions to the case of node-labeled and edge-weighted graphs. Besides, we make use of the *substructure fingerprint representation* [Fechner *et al.*, 2006] to define a class of kernels for protein binding sites. An experimental comparison of these graph kernels will be presented in Section 4 and discussed in Section 5.

2 Modeling Protein Binding Sites

To model protein binding sites as graphs, we build upon CavBase [Schmitt *et al.*, 2001, 2002], a database developed for the purpose of identifying and extracting putative

[*] These authors contributed equally to the work.

protein binding sites from structural data deposited in the protein database (PDB) [Berman *et al.*, 2000]. CavBase detects putative binding sites as cavities on the surface of proteins by using the LIGSITE algorithm [Hendlich *et al.*, 1997]. The geometry of a protein binding site is internally represented by a set of *pseudocenters*, spatial points that represent the physico-chemical properties of a surface patch within the binding site. Currently, CavBase uses seven types of pseudocenters (donor, acceptor, donor-acceptor, pi, aromatic, aliphatic and metal) that account for different types of possible interactions between residues of the binding site and the substrate of the protein. These pseudocenters are derived from the amino acid composition of the binding site.

As a natural way to model such structures, we make use of node-labeled and edge-weighted graphs. Nodes correspond to pseudocenters and are thus labeled with the pseudocenter type. On average, a graph representation of a binding pocket has around 100 nodes, though graphs with several hundred nodes and some extremes with thousands of nodes do exist.

Edges are weighted by the Euclidean distance between the pseudocenters and thus capture the geometry of the binding site. To reduce the complexity of the representation and increase algorithmic efficiency, we use an approximate representation in which edges exceeding a certain length are ignored; in this regard, a threshold of 11 Angström has proved to be a reasonable choice [Fober *et al.*, 2009]. Despite this approximation, our representation will produce graphs that are rather dense, as approximately 20 percent of all pairs of nodes are connected by an edge. Consequently, the graphs have a large number of cycles. Indeed, a cycle-free representation will normally not be able to reproduce the geometry of a binding site in an accurate way. As will be seen later on, this property leads to problems for certain types of kernel functions.

Formally, a node-labeled and edge-weighted graph will be denoted by $G = (V, E, l_V, l_E)$, where V is a finite set of nodes and $E \subseteq V \times V$ a set of edges. Moreover, $l_V : V \rightarrow \mathcal{L}_V$ is a function that maps each node to one among a finite set of labels $\mathcal{L}_V$. Likewise, $l_E : E \rightarrow \mathbb{R}_+$ is a mapping that assigns weights to edges. We define the size of a graph in terms of its number of nodes $|V|$. The adjacency matrix of a graph G will be denoted by A.

We note that, since our edges are undirected, it would be more correct to use a subset instead of a tuple representation. For convenience, however, we stick to the simpler tuple notation, with the implicit understanding that $(u, v) \in E$ implies $(v, u) \in E$ and $l_E((u, v)) = l_E((v, u))$.

3 Kernels for Node-Labeled and Edge-Weighted Graphs

Let $\mathcal{G}$ be a set of objects, in our case graphs. A $\mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}$ mapping k is called kernel if it is symmetric and positive definite, that is, $k(x, y) = k(y, x)$ for all $x, y \in \mathcal{G}$ and

$$\sum_{i,j=1}^{m} c_i c_j k(x_i, x_j) \geq 0$$

for all $m \in \mathbb{N}$, $\{c_1, \ldots, c_m\} \subseteq \mathbb{R}$, and $\{x_1, \ldots, x_m\} \subseteq \mathcal{G}$.

A generic way to define similarity measures for complex objects, such as graphs, is to use decomposition techniques, that is, to decompose a complex object into a set of simple substructures of a specific type, and to reduce the comparison to the level of these substructures. The idea is that, for such substructures, the definition of adequate similarity measures is less difficult and, hopefully, the computation more efficient. Therefore, graph kernels often belong to the class of *R-convolution kernels*, a special type of kernel especially suitable for composite objects in a discrete space. Generally, an R-convolution kernel $k : \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}$ can be expressed in the following from:

$$k(G, G') = \sum_{g \in R^{-1}(G)} \sum_{g' \in R^{-1}(G')} \kappa(g, g') \ , \qquad (1)$$

where $R^{-1}(G)$ denotes a decomposition of G into substructures, and κ is a kernel defined on such substructures. In the following, we consider specific instances of (1).

3.1 Random Walk Kernels

Random walk kernels were introduced in [Gärtner, 2003] for unweighted graphs. Roughly speaking, they decompose a graph into sequences of nodes generated by random walks, and count the number of identical random walks that can be found in two graphs. Thus, the random walk kernel is an R-convolution kernels with substructures given by paths. In the following, we present an extension of these kernels to the case of edge-weighted graphs.

Interestingly, to compute a graph kernel, it is not necessary to sample random walks. Instead, one can exploit an important property of the adjacency matrix A of a graph G, namely that $[A^n]_{i,j}$ is the number of paths of length n from node i to node j; here, A^n denotes the n-th power of A. Let $G_\times = G \times G'$ be the product graph of the graphs G and G', where the node and the edge set of $G_\times$ are defined as follows:

$$\begin{aligned} V_\times &= \{ (v_i, v'_j) \,|\, v_i \in V,\, v'_j \in V',\, l_V(v_i) = l_V(v'_j) \} \\ E_\times &= \{ ((v_i, v'_j), (v_k, v'_l)) \in V_\times \times V_\times \\ &\qquad | \ \|l_E(v_i, v_k) - l_E(v'_j, v'_l)\| \leq \epsilon \} \end{aligned}$$

Since $[A^n_\times]_{i,j}$ now corresponds to the number of equal paths of length n from node i to node j that occur in G as well as in G', the product graph $G_\times$ allows one to calculate $k(G, G')$ by performing simple matrix-operations. The requirement that node labels and edge weights have to match along two paths is implicitly encoded in the definition of the product graph (namely by the restriction to node pairs with $l_V(v_i) = l_V(v'_j)$ and edges with $\|l_E(v_i, v'_j) - l_E(v_k, v'_l)\| \leq \epsilon$); this idea was already used by [Borgwardt *et al.*, 2005], albeit only for discrete edge labels. The similarity of the graphs G and G', considering all equal paths of length 1 to ∞, is finally given by

$$k_{RW}(G, G') = \sum_{i,j=1}^{|V_\times|} \left[\sum_{k=0}^{\infty} \lambda_k \cdot A^k_\times \right]_{i,j} \ , \qquad (2)$$

where λ_k is a shrink factor that guarantees convergence of the series. For certain choices of λ, the above series can be calculated in a simple way. Choosing $\lambda_k = \lambda^k = (1/a)^k$, with $a \geq max_{v \in V_\times}\{\text{degree}(v)\}$, leads to the geometrical series, and (2) reduces to

$$k_{RW_{geo}}(G, G') = \sum_{i,j=1}^{|V_\times|} \left[(I - \lambda \cdot A_\times)^{-1} \right]_{i,j} \ . \qquad (3)$$

Choosing $\lambda_k = \frac{\beta^k}{k!}$ leads to the exponential series and to

$$k_{RW_{exp}}(G, G') = \sum_{i,j=1}^{|V_\times|} \left[e^{\beta \cdot A_\times} \right]_{i,j} \ .$$

Since the product graph is of quadratic size and matrix inversion has cubic complexity, the complexity of the random walk kernel is $\mathcal{O}(M^6)$, with $M = \max\{|V|, |V'|\}$.

3.2 Shortest Path Kernels

The random walk kernel considers an extremely large number of substructures (paths). Intuitively, this may not only come with a high computational complexity but also produce a certain redundancy. To reduce the number of substructures, Borgwardt [Borgwardt and Kriegel, 2005] proposed to consider only the shortest paths between two nodes, an idea which leads to the shortest path kernel. Again, we propose an extension of this kernel to the case of edge-weighted graphs.

For two nodes $v_i, v_j \in G$, let $sp(v_i, v_j)$ denote the length of the shortest path (sum of edge weights on the path) between these nodes, and let

$$SP(v_i, v_j) \; = \; (\{l_V(v_i), l_V(v_j)\}, sp(v_i, v_j)) \;\; .$$

Thus, a path is represented by its length and the labels of the start and the end node (while the node labels in-between are ignored). A simple kernel on substructures of this type is the identity (Dirac kernel):

$$\kappa_{path}(SP(v_i, v_j), \, SP(v_k, v_l)) \qquad (4)$$
$$= \begin{cases} 1 & \text{if } SP(v_i, v_j) = SP(v_k, v_l) \\ 0 & \text{else} \end{cases} \quad .$$

Since testing equality is of course not reasonable for real-valued edge lengths, we assume these lengths to be discretized (into bins of length $\delta = 1$).

Now, we can define the generalized shortest path kernel as follows:

$$k_{SP}(G, G') = \qquad (5)$$
$$\frac{1}{C} \sum_{v_i, v_j \in V} \; \sum_{v_k, v_l \in V'} \kappa_{path}(SP(v_i, v_j), SP(v_k, v_l)) \;\; ,$$

where $C = \frac{1}{4}(|V|^2 - |V|) \cdot (|V'|^2 - |V'|)$ is a normalizing factor that guarantees $0 \leq k_{SP}(G, G') \leq 1$.

To analyze the complexity of the shortest path kernel, assume $|V| = |V'| = M$. The computation of all shortest paths can be done using the Floyd-Warshall [Floyd, 1962] algorithm in time $\mathcal{O}(M^3)$. The results are stored in a shortest path matrix, in which the entry at position (i, j) gives the cost of the shortest path from node i to node j. We consider in a pairwise way all paths in both shortest path matrices and compare them using κ_{path} which needs time $\mathcal{O}(1)$. Since there are M^4 comparisons to perform, the shortest path kernel needs time $\mathcal{O}(M^4)$.

Representing a path only by its length and the labels of the start and end nodes and, correspondingly, using the Dirac kernel (4) for comparison does obviously come along with a considerable loss of information. To investigate whether performance can be improved by taking the labels of intermediate nodes into account, we developed another extension of the shortest path kernel. More specifically, we replaced the simple 0/1 measure (4) by a measure which compares the complete shortest paths sequence (sps). To this end, an sps $(v_1, v_2, \ldots, v_l)$ is represented in the form of a sequence

$$(l_V(v_1), l_E(v_1, v_2), l_V(v_2), \ldots, l_E(v_{n-1}, v_n), l_V(v_n))$$

in which node labels and (discretized) edge lengths occur alternately. To compare such sps, standard methods from sequence analysis can be used. A well-known approach based on the Levenshtein distance [Levenshtein, 1966] and dynamic programming has a runtime $\mathcal{O}(l_A \cdot l_B)$, where l_A and l_B is the length of sps A or B, respectively. When using appropriate scoring parameters like 1 for a match and 0 for a mismatch and for introducing a gap, this approach leads to a metric and therefore directly to a kernel.

To comply with the requirements of our application, the original dynamic programming approach to sequence alignment was modified as follows. First, recall that our sps involve two types of "symbols", namely node labels and edge weights. To ensure that the former are not aligned with the latter, which is obviously not reasonable, the cost for an assignment of this type was set to negative infinity. Second, note that long paths including many nodes represent more of the structure of a graph than paths of short length. Therefore, we normalize each score (similarity between two shortest paths) by dividing it by the length of the overall longest sps. This leads to an over-weighting of longer sequences since longer sequences are more likely to have higher scores than shorter sequences. In fact, we again obtain a measure with values between 0 and 1, which is used in (5) instead of (4).

In terms of runtime, the above extension of the shortest path kernel is of course very expensive. Again, we have to determine all shortest path, which can be done in time $\mathcal{O}(M^3)$. As explained above, the similarity between these paths is then measured using dynamic programming. Since the length of the sequences is $\mathcal{O}(M)$, and since there are $\mathcal{O}(M^2)$ sequences in both graphs, the total complexity for the pairwise comparison of all sps is $\mathcal{O}(M^3) + \mathcal{O}(M^2 \cdot M^2 \cdot M^2) = \mathcal{O}(M^6)$.

3.3 Fingerprint Kernels

A very simple type of kernel, which has nevertheless been applied successfully for learning on structured data such as molecular structures [Fechner *et al.*, 2006], is based on the idea of mapping a structured object to a fingerprint vector of fixed length and comparing these vectors afterward. Typically, each entry in this vector informs about the presence or absence of a specific substructure (pattern).

In our case, we consider as substructures all non-isomorphic graphs of size 3. Assuming n distinct node and k distinct edge labels, there exist

$$N(n, k) = \binom{n}{3} \cdot k^3 + n(n-1) \cdot k \cdot \binom{k+1}{2} + n \cdot \binom{k+2}{3}$$

substructures of this type, which can be verified by means of a case distinction: (i) All three node labels are distinct: There are $\binom{n}{3}$ possibilities to choose 3 distinct labels from a set of n labels. Moreover, since edges are ordered uniquely in this case, there exist k^3 possibilities for the edge labels. (ii) Two node labels are equal and different from the third: There are $n(n-1)$ possibilities to choose the two labels, one for the identically labeled nodes and one for the other. Assuming an arbitrary ordering on the nodes and edges, an isomorphism can switch the equally labeled nodes so that the ordering of two edges will change, too. To map isomorphic graphs uniquely, we sort the edges, which leads to only $k \cdot \binom{k+1}{2}$ possible edge combinations. (iii) All nodes have identical label: An isomorphism can reorder all nodes in this case. Therefore, to obtain a unique representation of the possible graphs, all edges must be sorted according to their label. Thus, there are n possible node labels and $\binom{k+2}{3}$ edge combinations.

For a graph G, let

$$f_G = \left(G \sqsupseteq t_1, G \sqsupseteq t_2, \ldots, G \sqsupseteq t_{N(n,k)}\right) \in \{0,1\}^{N(n,k)}$$

where $\{t_1, \ldots, t_{N(n,k)}\}$ is the set of all non-isomorphic subgraphs of size 3, numbered in an arbitrary but fixed order. The predicate $G \sqsupseteq t_i$ tests whether t_i is contained in G and, by convention, returns 1 if it evaluates to `true` and 0 otherwise. To compare two graphs G and G' in terms of their respective fingerprint vectors f_G and $f_{G'}$, different kernels can be used. The simplest approach is to look for the Hamming distance of the two vectors, which leads to

$$k_{FPH}(G, G') = \frac{1}{N(n,k)} \sum_{i=1}^{N(n,k)} \kappa_\delta([f_G]_i, [f_{G'}]_i) \quad , \tag{6}$$

where $[f_G]_i$ denotes the i-th entry in the vector f_G, and κ_δ is the Dirac kernel (i.e., $\kappa_\delta(x, y) = 1$ if $x = y$ and $= 0$ if $x \neq y$). As a potential disadvantage of this approach, note that it does not only reward the co-occurrence of a substructure in both graphs, but also the simultaneous absence: If the i-th pattern neither occurs in G nor in G', then $\kappa_\delta([f_G]_i, [f_{G'}]_i) = \kappa_\delta(0, 0) = 1$, which may not be desirable. An alternative measure avoiding this problem is the well-known Jaccard coefficient:

$$k_{FPJ}(G, G') = \frac{\sum_{i=1}^{N(n,k)} \min([f_G]_i, [f_{G'}]_i)}{\sum_{i=1}^{N(n,k)} \max([f_G]_i, [f_{G'}]_i)} \quad . \tag{7}$$

Our current implementation of the fingerprint approach is a naive one, in which testing the presence of a substructure in a graph G has complexity $O(M^3)$, with $M = |V|$ the number of nodes in G. Thus, the overall complexity of computing $k(G, G')$ is $O(N(n,k) \cdot M^3)$, with $M = \max(|V|, |V'|)$. However, we can utilize the fact that we can abort the search for a substructure as soon as we have found the first occurence.

3.4 Kernels based on Fuzzy Fingerprints

The discretization of edge weights needed for the previous approach can be criticized for several reasons. Some of the disadvantages, notably the abrupt transition between the presence and absence of a subgraph due to a very small change of an edge length, can be avoided by means of a *fuzzy* discretization. A fuzzy partition of a domain X (in our case $\mathbb{R}_+$) is defined by a finite family of fuzzy subsets $F_1, F_2, \ldots, F_k$ of X such that $\sum_{i=1}^k F_i(x) > 0$ for all $x \in X$; typically, one even requires that $\sum_{i=1}^k F_i(x) = 1$ for all $x \in X$. Concretely, we shall use a fuzzy partition in which F_i is defined by

$$F_i(x) = \max\{0, 1 - |x - i|\} \quad .$$

Thus, F_i can be interpreted as the fuzzy subset of numbers "approximately equal to i".

A pattern t is now a graph of size 3 whose nodes are labeled as before, but whose edges are labeled with fuzzy numbers of the form F_i. A subgraph S of a graph G with real-valued edge lengths can be isomorphic to a pattern t to a certain degree. Let a_i be the label of the i-th node in S, and x_{ij} the length of the edge between node i and node j. Likewise, let b_i be the label of the i-th node in t, and F_{ij} the length of the edge between node i and node j. The degree of isomorphism of t and S, denoted $[t \sim S]$, is then given by

$$\max_{\pi \in S_3} \begin{cases} \min\{F_{12}(y_{12}), F_{13}(y_{13}), F_{23}(y_{23})\} & \text{if } M(\pi) \\ 0 & \text{otherwise} \end{cases}$$

where $y_{ij} = x_{\pi(i),\pi(j)}$, S_3 is the set of all permutation $\{1, 2, 3\} \to \{1, 2, 3\}$, and $M(\pi)$ is true if

$$(a_1 = b_{\pi(1)}) \wedge (a_2 = b_{\pi(2)}) \wedge (a_3 = b_{\pi(3)})$$

and false otherwise. Likewise, the degree to which t is present in the graph G is then given by

$$[G \sqsupseteq t] = \max_S \, [t \sim S] \quad , \tag{8}$$

where the maximum is taken over all subgraphs of size 3 in G. This value defines the entry for the pattern t in the (fuzzy) fingerprint vector $[f_G]$ of G.

In the non-fuzzy approach, the search for a pattern t in a graph G can be stopped as soon as the pattern has been found. Here, this is no longer possible, since the maximum (8) has to be determined. To accelerate the calculation of this maximum, we make use of a canonical form describing graphs of size 3. To this end, we distinguish three cases:

- All node labels are equal. In this case, the canonical form is given by the node label followed by the edge lengths in increasing order.
- Two nodes have an identical label. The canonical form starts with the node label that appears once in the graph followed by the label that appears twice, the edge weight between the nodes with the same label, and finally the remaining two edge weights in increasing order.
- All nodes have different labels. The canonical form is then defined by the three occurring labels, sorted in a lexicographic order, the edge length between the first and the second, the second and the third, and finally the first and the third node.

All three cases are illustrated by an example in Fig. 3.4.

We denote the set of all canonical forms by Σ. The above representation enables the definition of a bijective function $i : \Sigma \to \{1, \ldots, N(n,k)\} \subset \mathbb{N}$ assigning a unique number to each each form and, therefore, subgraph of size 3. Using this mapping, the calculation of the fingerprint vector for a graph $G = (V, E)$ can be done in a more efficient way. Instead of counting, for each entry i in the fingerprint vector, how often subgraph g_i appears in G, we can enumerate all subgraphs of size 3 in G. For each subgraph g_i of G, we perform the transformation to its canonical form σ_i (in time $\mathcal{O}(1)$), evaluate the function $i(\sigma_i)$ to determine the position of g_i in the fingerprint vector (in time $\mathcal{O}(1)$), and finally update the entry at this position in the vector. Doing this for all $\binom{M}{3} = \mathcal{O}(M^3)$ subgraphs of size 3 leads to a runtime of $\mathcal{O}(M^3)$ in comparison to $\mathcal{O}(M^3) \cdot N(n,k)$, where $N(n,k)$ is usually a large number (in our case 36,729).

The fingerprint vectors eventually obtained are "fuzzy" in the sense of having entries in the unit interval $[0, 1]$ instead of being either 0 or 1. From a machine learning point of view, this is interesting as it may increase discriminatory power. To compare the vectors, we again use the approach based on the Jaccard coefficient in (7), where we substitute the logical operators by a t-norm and t-conorm, respectively:

$$k_{FFP}(G, G') = \frac{\sum_{i=1}^{N(n,k)} \top([f_G]_i, [f_{G'}]_i)}{\sum_{i=1}^{N(n,k)} \bot([f_G]_i, [f_{G'}]_i)} \quad .$$

For the experiments we used $\top(a, b) = \min(a, b)$ and $\bot(a, b) = \max(a, b)$.

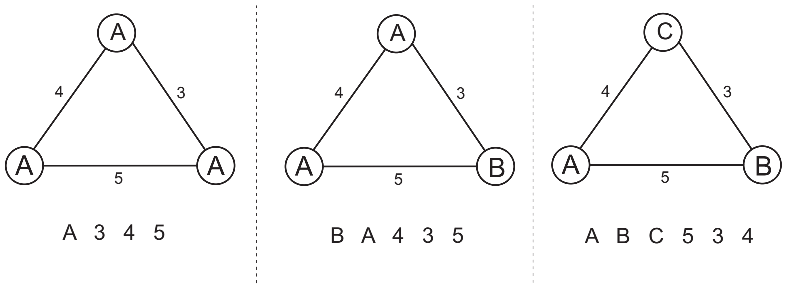


Figure 1: The three possible cases that can occur: all labels identical, two labels identical and all labels unique.

4 Experimental Evaluation

In our experiments, we compared the graph kernels discussed in the previous section, namely the random walk kernel (RW) using (3) with a given by the maximum size of the graphs in the data set (plus 1), the shortest path kernel (SP) and its extended version based on sequence alignment (SPSA), the fingerprint kernel based on (6) and (7), respectively (FPH and FPJ), and the fuzzy fingerprint-kernel (FFP). Moreover, to get an idea of their absolute performance, we additionally included two state-of-the-art methods for comparing protein binding sites in terms of their similarity. Both approaches are based on the concept of a *graph alignment* that has been introduced in [Weskamp *et al.*, 2007]. The first method (GA) is the original algorithm proposed in the same paper, which is based on a heuristic (greedy) optimization strategy. The second method (GAVEO) makes use of evolutionary optimization techniques to compute a graph alignment [Fober *et al.*, 2009]. Both methods need a number of parameters, which we defined as recommended in [Weskamp *et al.*, 2007]. For the kernel methods, we set the parameter ϵ (tolerance for edge length comparison) to 0.2.

The assessment of a similarity measure for molecular structures, such as protein binding sites, is clearly a nontrivial problem. In particular, since the concept of similarity by itself is rather vague and subjective, it is difficult to evaluate corresponding measures in an objective way. To circumvent this problem, we propose to evaluate similarity measures in an indirect way, namely by means of their performance in the context of nearest neighbor (NN) classification. The underlying idea is that, the better a similarity measure is, the better he predictive performance we expect from an NN classifier using this measure for determining similar cases.

4.1 Data

We selected two classes of binding sites that bind to NADH or ATP, respectively. This gives rise to a binary classification problem: Given a protein binding site, predict whether it binds NADH or ATP. More concretely, we compiled a set of 355 protein binding pockets representing two classes of proteins that share, respectively, ATP and NADH as a cofactor. To this end, we used CavBase to retrieve all known ATP and NADH binding pockets that were co-crystallized with the respective ligand. Subsequently, we reduced the set to one cavity per protein, thus representing the enzymes by a single binding pocket. As protein ligands adopt different conformations due to their structural flexibility, it is likely that the ligands in our data set are bound in completely different conformations, hence the corresponding binding pockets do not necessarily share much structural similarity. To ensure a minimum level of similarity, we therefore utilized the ligand information available for these binding pockets, as these structures where all cocrystallized with the corresponding ligand. Using the Kabsch algorithm [Kabsch, 1976], we calculated the root mean squared deviation (RMSD) between pairs of ligand structures and combined all proteins whose ligands yielded a RMSD value below a threshold of 0.4, thus ensuring that the ligands are roughly oriented in the same way. This value was chosen as a trade-off between data set size and similarity. Eventually, we thus obtained a two-class data set comprising 214 NADH-binding proteins and 141 ATP-binding proteins.

4.2 Results

The performance of the different methods, using an SVM (LIBSVM implementation) and a simple k-nearest neighbor classifier ($k = 1, 3, 5, 7, 9$) for prediction, is summarized in Table 1. Since all methods obviously fulfill the kernel properties with the exception of GA and GAVEO, these methods can be used as precomputed kernel functions for the LIBSVM package.

Table 2 shows the average time complexity of the methods, namely the time needed for a single pairwise comparison of two structures. These numbers have been determined by averaging over 1000 comparisons with randomly chosen structures. FFP_{index} gives the runtime of the fuzzy-fingerprint kernel that uses the index-function described above. Note the significant improvement regarding runtime in comparison to the fuzzy-fingerprint kernel without the index function.

5 Discussion and Conclusion

The results convey are relatively clear picture: The fingerprint kernels perform best, the random walk and shortest path kernel worst, and the graph alignment methods are in-between. The overall best results are achieved by the Jaccard-variant of the fuzzy fingerprint kernel. In terms of efficiency, the fingerprint kernels are superior, too. Thus, this type of kernel is clearly of high interest in the context of comparing protein binding sites.

The poor performance of the random walk and shortest path kernels is a bit astonishing at first sight. Our extension of the shortest path kernel based on sequence comparison yield some improvement when using SVM as a classifier, though it is still not competitive with the fingerprint kernels. The failing of these types of kernel can possibly be attributed to their characteristics as R-convolution kernels. In general, the ‘all-against-all’ comparison of substructures performed by kernels of this type appears to be problematic for large graphs or, more generally, for diverse objects consisting of many substructures. It leads to a kind of averaging effect, and indeed, we observed that the entries in

Table 1: Classification rates of an SVM and a k-nearest-neighbor classifier in a leave-one-out cross validation using different values of k and different similarity measures: random walk kernel (RW), shortest path kernel (SP), shortest path kernel based on sequence alignment (SPSA), fingerprint kernel (FPH, FPJ), fuzzy fingerprint kernel (FFP), and graph alignment (GA, GAVEO).

method	RW	SP	SPSA	FPH	FPJ	FFP	GA	GAVEO
SVM	0.606	0.625	0.707	0.916	0.907	0.916	—	—
k = 1	0.597	0.606	0.620	0.828	0.842	0.879	0.766	0.789
k = 3	0.597	0.628	0.546	0.839	0.882	0.887	0.718	0.766
k = 5	0.597	0.634	0.552	0.839	0.873	0.887	0.724	0.780
k = 7	0.608	0.625	0.566	0.819	0.859	0.854	0.718	0.786
k = 9	0.608	0.634	0.597	0.814	0.836	0.839	0.713	0.766

Table 2: Average runtime and standard deviation (in seconds) of the different methods for a single pairwise comparison.

method	RW	SP	SPSA	FP
runtime	65.51 ± 89.07	9.75 ± 97.77	574.43 ± 4089.70	2.05 ± 3.66

method	FFP	FFP$_{index}$	GA	GAVEO
runtime	53.99 ± 121.01	17.90 ± 66.34	121.74 ± 418.02	> 5 min

our kernel matrix are all very similar. Moreover, in the random walk kernel, nodes and edges can appear more than once in a random walk, a problem known as *tottering*. This problem becomes especially severe in the presence of many cycles within a graph, a property which, as mentioned earlier, our graph descriptors of protein binding sites will inevitably exhibit. The shortest path kernel avoids tottering but has another problem known as *halting*: As it only looks at shortest paths, it tends to be dominated by a large number of paths with very few nodes. As we consider graphs representing geometric constraints within a binding pocket, this is likely to result in a loss of information.

The strong performance of the fingerprint kernel suggests to elaborate on this approach in more detail. In fact, the approach presented in this paper is rather simple and can be extended in different ways. First, substructures other than subgraphs of size 3 might be considered, even though our experience so far has shown that this class of patterns is able to capture considerable information while still being manageable in terms of complexity. Second, the fingerprint vectors could be constructed (and compared) in a more sophisticated way. For example, instead of just indicating the presence or absence of a pattern, one may count its number of occurrences and then apply similarity measures for frequency vectors.

References

Johannes Berg and Michael Lässig. Local graph alignment and motif search in biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(41):14689–14694, 2004.

H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, , and P. E. Bourne. The protein data bank. *Nucleic Acids Research*, 28:235–242, 2000.

K. M. Borgwardt and H. P. Kriegel. Shortest-path kernels on graphs. In *International Conference on Data Mining*, pages 74–81, Houston, Texas, 2005.

Karsten Borgwardt, Cheng Soon Ong, Stefan Schönauer, S. V. N. Vishwanathan, Alex J. Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(21):i47 – i56, 2005.

K. M. Borgwardt. *Graph Kernels*. PhD thesis, Ludwig-Maximilians-Universität München, Germany, 2007.

Horst Bunke and Xiaoyi Jiang. Graph matching and similarity. *Intelligent systems and interfaces*, 15:281 – 304, 2000.

N. Fechner, G. Hinselmann, and A. Zell. Implicitly defined substructure fingerprints for support vector machines. In *German Conference on Chemoinformatics*, 2006.

R. W. Floyd. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345, 1962.

Thomas Fober, Marco Mernberger, Gerhard Klebe, and Eyke Hüllermeier. Evolutionary construction of multiple graph alignments for the structural analysis of biomolecules. *Bioinformatics*, 2009.

Thomas Gärtner. A survey of kernels for structured data. *SIGKKD Explorations*, 5(1):49 – 58, 2003.

J. F. Gibrat, T. Madej, and S. H. Bryant. Surprising similarities in structure comparison. *Current Opinion in Structural Biology*, 6(3):377–385, 1996.

M. Hendlich, F. Rippmann, and G. Barnickel. LIGSITE: Automatic and efficient detection of potential small molecule-binding sites in proteins. *Journal of Molecular Graphics and Modelling*, 15:359–363, 1997.

M. Jambon, A. Imberty, G. Deleage, and C. Geourjon. A New Bioinformatic Approach to Detect Common 3 D Sites in Protein Structures. *Proteins Structure Function and Genetics*, 52(2):137–145, 2003.

Wolfgang Kabsch. A solution of the best rotation to relate two sets of vectors. *Acta Crystallographica*, 32:922–923, 1976.

V. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

Michael Neuhaus and Horst Bunke. *Briding the Gap between Graph Edit Distance and Kernel Machines*. World Scientific, New Jersey, 2007.

J. Raymond and P. Willett. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *Journal of Computer-Aided Molecular Design*, 16(7):521–533, 2002.

J.W. Raymond, E.J. Gardiner, and P. Willett. Heuristics for Similarity Searching of Chemical Graphs Using a Maximum Common Edge Subgraph Algorithm. *Jorunal of Chemical Information and Computer Sciences*, 42(2):305–316, 2002.

S. Schmitt, M. Hendlich, and G. Klebe. From structure to function: A new approach to detect functional similarity among proteins independent from sequence and fold homology. *Angewandte Chemie International Edition*, 40(17):3141 – 3144, 2001.

S. Schmitt, D. Kuhn, and G. Klebe. A new method to detect related function among proteins independent of sequence and fold homology. *Journal of Molecular Biology*, 323(2):387–406, 2002.

N. Weskamp, E. Hüllermeier, D. Kuhn, and G. Klebe. Multiple graph alignment for the structural analysis of protein active sites. *IEEE Transactions on Computational Biology and Bioinformatics*, 4(2):310–320, 2007.

Relation Extraction for Monitoring Economic Networks (Resubmission)

Martin Had, Felix Jungermann and Katharina Morik
Technical University of Dortmund
Department of Computer Science - Artificial Intelligence Group
Baroper Strasse 301, 44227 Dortmund, Germany

Abstract

Relation extraction from texts is a research topic since the message understanding conferences. Most investigations dealt with English texts. However, the heuristics found for these do not perform well when applied to a language with free word order, as is, e.g., German. In this paper, we present a German annotated corpus for *relation extraction*. We have implemented the state of the art methods of *relation extraction* using kernel methods and evaluate them on this corpus. The poor results led to a feature set which focusses on all words of the sentence and a tree kernel which includes words, in addition to the syntactic structure. The *relation extraction* is applied to monitoring a graph of economic company-directors network.

1 Introduction

Social networks have raised scientific attention, the goals ranging from enhancing recommender systems [Debnath *et al.*, 2008; Palau *et al.*, 2004; Domingos and Richardson, 2001] to gaining scientific insights [Golder *et al.*, 2007; Zhou *et al.*, 2007a]. Where the taggings, mailings, co-authorship, or citations in communities have well been investigated, the economic relationships between companies and their networking have less been studied.

Today's search engines are not prepared to answer questions like "show me all companies that have merged with Volkswagen". In order to get that information anyway, it would be necessary to do an extensive search and consider several sources. This is time consuming and tedious. This is why question answering approaches require automatic *relation extraction*.

Moreover, it is important to represent the extracted information in a compact and easy to access manner. Especially concerning *relation extraction*, the extracted entities and relations can be represented using an (un-)directed graph.

In this paper, we present an approach to monitoring economic information in the world wide web using a graph-based representation. We will show that it is possible to extract additional information using *relation extraction* techniques, which have not yet successfully been used on German texts, because German language features problems, which other languages – especially English – do not face. A comparison of our feature set and enhanced tree kernel with state of the art methods illustrates the importance of a balanced use of semantic and syntactic information. First, we describe the state of the art in *relation extraction* using kernel methods, then we present our application, before we introduce or enhancement of the method and the experimental results.

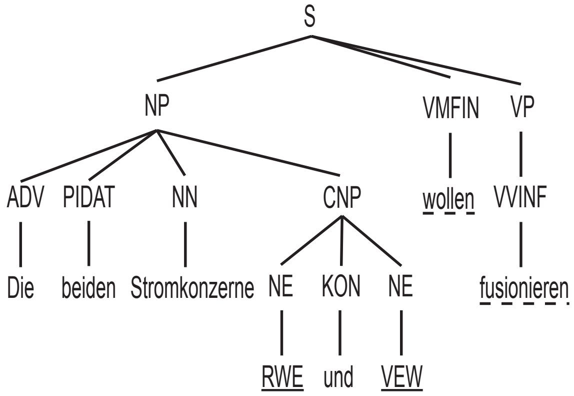


Figure 1: A parse tree for a German sentence containing a merger-relation.

2 Kernel Methods for Relation Extraction

The ACE RDC task [Lin, 2004] defines a relation as a valid combination of two entities that are mentioned in the same sentence and have a connection to each other. Relations may be symmetric or asymmetric. The schema of i relations in a sentence s is defined as follows:

Definition 1 Relation candidates *in a sentence:*

$$R_i(Sentence\ s) := \langle Type_m \in relationtypes, (Argument_1, Argument_2)\rangle$$

where m is one relationtype of all the possible $relationtypes$, $(Argument_1 \neq Argument_2) \in entities_s$ and $entities_s$ is the set of entities contained in the current sentence.

Structured information of a sentence e.g. is the syntactic parse tree (an example can be seen in Figure 1), where each node follows a grammar production rule.

By splitting up a tree in subtrees (see Figure 2) it is possible to calculate the similarity of two trees by counting their common subtrees. The set of subtrees of a parse tree consists of every substructure that can be built by applying the grammatical rule set of the original tree.

2.1 Linear Kernels

First experiments on *relation extraction* have been done by just using feature-based methods. That made it necessary to manually create a large set of 'flat' features describing

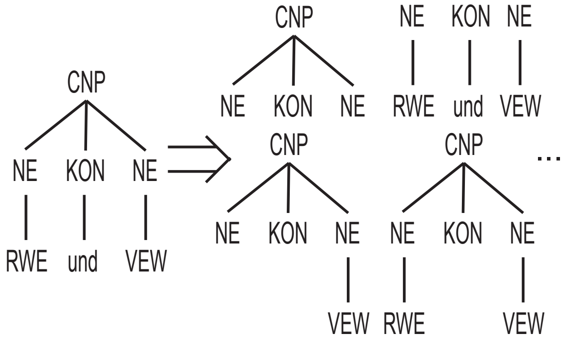


Figure 2: Some subtrees of a tree

the relation and comparing the similarities of these feature-vectors in order to find the best discriminating classification function. The most efficient way to compare feature-vectors is based on kernel functions which can be embedded in various machine-learning algorithms like support vector machines or clustering methods. A linear kernel on feature-vectors, x and z, is defined as their inner product:

Definition 2 *A* linear kernel*:*

$$K(x,z) = \sum_n \phi_n(x)\phi_n(z) \tag{1}$$

where $\phi_n(x)$ is the n-th feature of x.

2.2 Convolution Kernels

Converting syntactic structures into feature-vectors is tedious [Zhao and Grishman, 2005] [Zhou *et al.*, 2005]. This overhead is avoided when using a kernel function, which operates on any discrete structure [Haussler, 1999]. Because of the formulation as a kernel, the calculation of the inner product requires the enumeration of substructures only implicitly.

Definition 3 *Let $x \in X$ be a composite structure and $\vec{x} = x_1, \ldots, x_p$ are its parts, where each $x_i \in X_i$ for $i = 1, \ldots, p$ and all X_i are countable sets. The relation $R(\vec{x}, x)$ is true, iff $x_1, \ldots, x_p$ are all parts of x. As a special case, X being the set of all p-degree ordered, rooted trees and $X_1 = \cdots = X_p = X$, the relation R can be used iteratively to define more complex structures in X.*

Given $x, z \in X$ and $\vec{x} = x_1, \ldots, x_p$, $\vec{z} = z_1, \ldots, z_p$ and a kernel K_i for X_i measuring the similarity $K_i(x_i, z_i)$, then the similarity $K(x,z)$ is defined as the following generalized convolution

$$K(x,z) = \sum_{\{\vec{x}|R(\vec{x},x)\}} \sum_{\{\vec{z}|R(\vec{z},z)\}} \prod_{i=1}^{p} K_i(x_i, z_i) \tag{2}$$

[Haussler, 1999]p.5f

Convolution kernels characterize the similarity of parse trees by the similarity of their subtrees [Collins and Duffy, 2001]. Within the kernel calculation, all subtrees of the trees are compared. They are (implicitly) represented as a vector:

$$\Phi(T) = (subtree_1(T), \ldots, subtree_m(T)) \tag{3}$$

where $subtree_i$ means the number of occurrences of the i-th subtree in T. The number of common subtrees is summed up. The worst case runtime is $O(|N_1| \times |N_2|)$, being N_t the set of nodes of a tree T_t.

Definition 4 *The* tree kernel *computes a scalar product:*

$$K(T_1, T_2) = \langle \mathbf{h}(T_1), \mathbf{h}(T_2) \rangle \tag{4}$$

$$h_i(T_1) = \sum_{n_1 \in N_1} I_i(n_1) \tag{5}$$

where the indicator function I_i is defined for the nodes n_1 in N_1 of T_1 and n_2 in N_2 for T_2 as 1, iff the $i-$th subtree is rooted in node n, otherwise I_i is defined as 0. Hence,

$$K(T_1, T_2) = \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} \sum_i I_i(n_1) I_i(n_2) \tag{6}$$

$$= \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} \Delta(n_1, n_2) \tag{7}$$

The calculation of Δ is done recursively by following three simple rules:

- If the grammar production rules of n_1 and n_2 are different: $\Delta(n_1, n_2) = 0$
- If the production rules in n_1 and n_2 are equal and n_1 and n_2 are pre-terminals (last node before a leaf): $\Delta(n_1, n_2) = \lambda$
- If the production rules in n_1 are n_2 equal and n_1 and n_2 are non pre-terminals:

$$\Delta(n_1, n_2) = \lambda \prod_{j=1}^{nc(n_1)} (1 + \Delta(ch(n_1, j), ch(n_2, j))) \tag{8}$$

$nc(n_1) =$ number of children of node n_1
$ch(n_1, i) = i$th child of node n_1
$\lambda =$ parameter to downweight the contribution of large tree fragments exponentially with their size.

[Moschitti, 2006] designed an algorithm for the above calculation that has linear runtime on average due to a clever preprocessing step. Nodes that don't need to be considered by the kernel are filtered out by sorting and comparing the production rules of both trees in advance ("Fast Tree Kernel" *FTK*).

[Zhou *et al.*, 2007b] extended the FTK kernel to become context sensitive by looking back at the path above the ancestors of the root node of each subtree. The left side of the production rule is taking into account $m - 1$ steps towards the root. The kernel calculation itself sums up the calculations for each set of production rules created for $1 \ldots m$. In the special case $m = 1$ the kernel result is the same as with the non context-sensitive kernel.

$$K_C(T_1, T_2) = \sum_{i=1}^{m} \sum_{n_1^i[1] \in N_i^1[1], n_1^i[2] \in N_i^1[2]} \Delta(n_1^i[1], n_1^i[2]) \tag{9}$$

- m the number of ancestor nodes to consider.
- $n_1^i[j]$ is a node of tree j with a production rule over i ancestors. $n_1[j]$ is the root node of the context free subtree, the ancestor node of $n_k[j]$ is $n_{k+1}[j]$.
- $N_1^i[j]$ is the set of all nodes with their production rules over i ancestor.

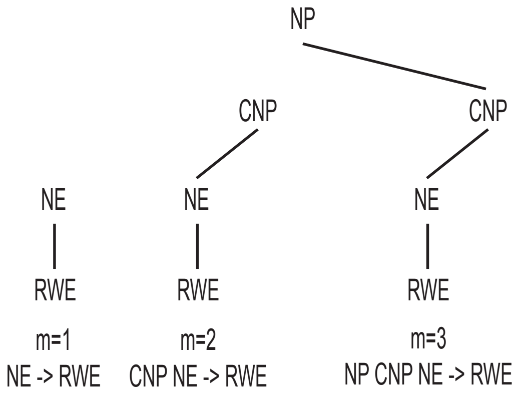


Figure 3: Production rules of subtree root node *(*NE) for different m values

2.3 Composite Kernels

[Zhang *et al.*, 2006] showed that better results can be achieved with a combination of linear and tree kernels. [Haussler, 1999] showed that the class of kernels is closed under product and addition. This implies that combining two kernels is possible and results in a new kernel which is called a *composite kernel*, defined as follows:

$$K(x, z) = K_1(x, z) \circ K_2(x, z) \tag{10}$$

In the case of *relation extraction* the composite kernel combines a linear and a convolution tree kernel. [Zhang *et al.*, 2006] propose a linear combination (11) and a polynomial expansion (12).

$$K(x, z) = \alpha \cdot \hat{K}_L(x, z) + (1 - \alpha) \cdot \hat{K}_T(x, z) \tag{11}$$

$$K(x, z) = \alpha \cdot \hat{K}_L^P(x, z) + (1 - \alpha) \cdot \hat{K}_T(x, z) \tag{12}$$

where x and z are *relation candidates* that consist of flat features and structured information, each kernel is given the right input for its kind. $K_L^P(x, z)$ is the polynomial expansion of $K_L(x, z)$ with degree $d = 2$ i.e. $K_L^P(x, z) = (K_L(x, z) + 1)^2$. By setting the α value the influence of each kernel type can be adjusted. Both kernels are normalized in kernel space before the combination:

$$K'(T_1, T_2) = \frac{K(T_1, T_2)}{\sqrt{K(T_1, T_2)K(T_1, T_2)}} \tag{13}$$

2.4 State-of-the-art composite kernels for Relation Extraction

In addition to just using the composite kernel on the full parse tree of a sentence, [Zhang *et al.*, 2006] examined several ways of pruning the parse tree in order to get differently shaped subtrees on which the treekernel performs as well or better as on the full tree. They showed that the shortest path-enclosed tree (SPT) which is the minimal subtree containing the two entities of a *relation candidate* performs best for the ACE 2003 RDC corpus.

But [Zhou *et al.*, 2007b] showed that the ACE corpus contains relations for which the SPT is not sufficient. These relations are indicated by their related verb. Figure 1 shows a relation of our corpus which is indicated by its related verb, too. But in contrast to the ACE corpus which just contains a few relations of this type, our corpus has many. The type of relation and the specialty of the German language are responsible for this fact.

The strict and binary decisions of the tree kernel are the main disadvantage of this method. [Zhou *et al.*, 2007b] tried to overcome this problem by embedding syntactic features into the parse tree directly above the leaf-nodes. Moreover, syntactic structure is already covered by the tree kernel, adding it in terms of features does not help generalization. [Zhang *et al.*, 2007] generalized the production rules of the parse tree in order to achieve better performance. The strict decisions of a convolution tree kernel (remind Section 2.2) make the kernel returning 'unequal' confronted with two production rules "$NP \rightarrow Det\ Adj\ N$" and "$NP \rightarrow Det\ N$" although they might contain similar terminals ("$NP \rightarrow$ a red car" and "$NP \rightarrow$ a car"). To avoid such behavior they proposed inserting of optional nodes into production rules to generalize them. Additionally similar part of speech tags in the parse tree can be processed in an equal way – multiplied with a penalty term.

This is a step into the right direction. However, only syntactic variance is handled. Since words carry most of the semantic information, moving them into the tree kernel could well help to generalize in a more semantic way.

Related kernels for Relation Extraction

There are several related approaches for *relation extraction* differing from the ones already presented. [Vishwanathan and Smola, 2002] presented a general kernel function for trees and its subtrees. [Zelenko *et al.*, 2003] used a kernel function on shallow parse trees. Bunescu and Mooney used a kernel function on the shortest path between two entities in a dependency tree [Buncescu and Mooney, 2005]. Additionally they used the context of entities for *relation extraction* [Bunescu and Mooney, 2006].

3 Monitoring the merger event in an economic network

The enhancement of the state of the art in *relation extraction* which is described in Section 4 became necessary when we developed the economic network based on German sources. We did not want to manually build and update its database. Extracting relations from documents directly allows to automatically accomplish the data about companies and their board members with relationships between them. Hence, the network can be monitored and is always up-to-date.

3.1 Building-up the economic network

Building up the economic network starts with extracting companies and their board of directors. The extraction of the named entities "company" and "board member" is quite simple, because there exist several web archives of companies which are semi-structured. Hence, companies and their representatives can easily be extracted using simple regular expressions. The initial stock of data is stored in an SQL database. It consists of about 2,000 different big companies from throughout the world. Basic information includes only address and industry but most entries provide a lot more details about members of the board of directorate, share ownership, shareholding and some key performance indicators. Many of these companies (here: 1,354) are connected to one other company at least, by sharing a member of the directorate. The best connected company even has 37 different outgoing directorate connections. These numbers support the assumption, that the graph built from these relations can reveal significant structures in the business world.

From the SQL data base, a network $G = (V, E)$ is built containing entities ($v \in V$) and relations ($e \in E$) between

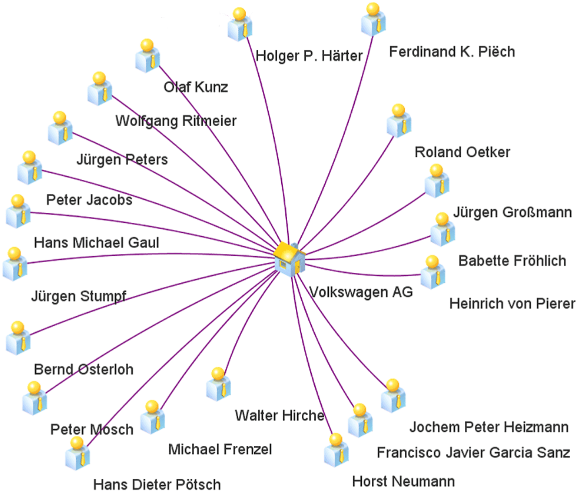


Figure 4: Selecting Volkswagen AG (VW) from all companies, the involved responsible persons are displayed.

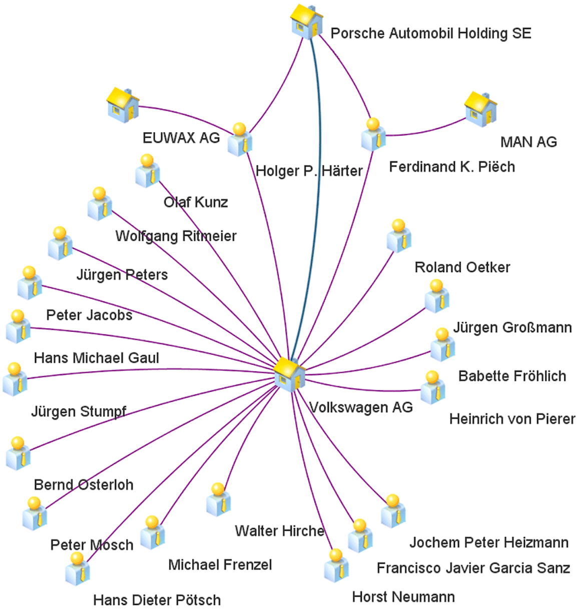


Figure 5: Two directors of the board of VW are directors of Porsche, as well. The merger-relation holds between VW and Porsche (indicated by a blue line between the companies).

entities. Its visualization is performed using the JUNG-Framework [O'Madadhain *et al.*, 2003]. The human-computer interface allows users to select a company and move to the involved persons, from which the user may move to all the companies in which they play a role – thus browsing through the basic social network graph of economy. Figure 4 shows an example. Since the archives do not change their structure whenever the content changes, the database is easily updated.

3.2 Extracting the merger-relation from web documents

For monitoring the web of economy, the merger-relation is most interesting. A company merger-relation is defined as a symmetric relation between two different companies becoming one. The *relation extraction* is restricted to those companies expressed within one sentence. Exemplary German sentences containing a (negative or positive) merger-relation are:

- Porsche hat die Pläne einer Übernahme von VW aufgegeben. (negative – Porsche abandoned the plans to take over VW.)
- Auch ein Zusammenschluss von Commerzbank und Dresdner Bank – die heute zur Münchener Allianz gehört – scheiterte. (negative – A fusion of Commerzbank and Dresdner Bank – belonging to the Münchener Allianz – failed.)
- Wie Air Berlin jetzt mitteilte, übernimmt der Billigflieger den Konkurrenten dba. (positive – The low-cost carrier is taking over dba, as Air Berlin informed.)

To get a preselection of relevant documents the web is crawled for information about the 30 DAX indexed German companies. Given a list of known company names, the texts of the resulting websites are tagged in the IOB-scheme indicating "company"-entities. Only those sentences containing at least two company entities are selected for further processing. It is then the task of *relation extraction* to identify the true merger-relations between two companies. Of course, simple co-occurrence is not sufficient for this task. Note, that a sentence with three company names can include none, one, or two merger-relations. Hence, we applied our method described in Section 4. Details on the experiments are given in Section 5. Figure 5 shows an example of a found merger-relation.

4 Relation extraction with an enhanced composite kernel

We have implemented the state-of-the-art kernel method in Java, extending the kernel functions of SVM^{light} [Joachims, 1989]. We also have developed an information extraction plug-in [Jungermann, 2009] for RapidMiner (formerly Yale) [Mierswa *et al.*, 2006] including the composite kernel and all necessary preprocessing. When handing over the examples to the kernel functions, an example is split into the features for the linear kernel and into the tree for the tree kernel. When passing the tree to the kernel function, it may be pruned and enriched by new features.

We changed two aspects concerning the state-of-the-art composite kernels used for *relation extraction*:

- First, we widened the featureset used for the linear kernel.
- Second, we added semantic information to the tree kernel.

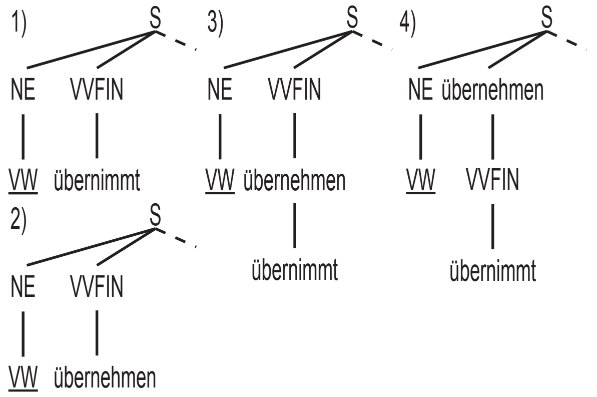


Figure 6: Word stems at different depth-levels in the parse tree

Features which contain words or word-parts of special positions in the sentence related to the entity's position showed to be useful for named entity recognition. However, for *relation extraction*, the position is no longer decisive. The information about the relation is spread all over the sentences, shows up at very different places, and can, hence, not be generalized. The clue verb *fusionieren*, for instance, may occur at various positions of the word sequence (see Figure 1). Especially for distinguishing between positive and negative *relation candidates*, the contextual information is not restricted to the words between the entities or to some words in front or behind the entities, as assumed by the feature set in [Zhou *et al.*, 2005]). In order to capture the influence of words that can act as an indicator for a relation we extract the word vector (containing just the word stems) of the complete sentence and add it to the linear features. In a separate experiment setting we use the features presented by [Zhou *et al.*, 2005], for comparison.

The second of our enhancements concerns the tree kernel. Figure 1 shows a parse tree of our corpus containing two entities (underlined solid) and the merger-indicating verbs (underlined dashed). It is easy to see that well-known subtrees for better *relation extraction* like shortest path-enclosing trees (SPT) will not work well in this context. But using the whole and unaltered parse tree will not work as well. The reason is, if a sentence contains positive and negative *relation candidates* the identical parse tree would be used for both *relation candidate*-types.

Using the context-sensitive parse tree of [Zhou *et al.*, 2007b] is promising. But this approach needs well-trained parsers which are still not available in an appropriate version for the German language. We therefore generalized the parse tree by adding syntactical information directly into the tree. First of all we marked the entities of the current *relation candidate* in the corresponding parse tree. In addition, we added semantic information into the parse tree by introducing extra nodes containing the word stems of the sentence at different depths.

Figure 6 shows four different types of parse trees used in our experimental settings. The first one (1)) is the original parse tree. In parse tree 2) we have replaced all terminals by their stems. Parse tree 3) is the tree after inserting the stem at depth 0. The depth is the depth of the stem in relation to the depth of the pre-terminal symbol. 4) is the tree after inserting the stem at depth 1. The word 'VW' has no stem, so nothing is inserted.

5 Experiments

Our companies corpus consists of 1,698 sentences containing 3,602 *relation candidates*. 2,930 of these *relation candidates* are negative ones (being no merger-relation), and 672 *relation candidates* are true merger-relations. Only 98 of these 1,689 sentences contain multiple *relation candidates* with different labels. Compared to other *relation extraction* datasets this distribution is very skewed and leads to the behavior described in the following Section. The ACE04 corpus for instance contains 2,981 sentences out of which 1,654 sentences contain at least a true relation and a negative candidate.

We produced several training sets with different attribute sets to compare our enhancements with the state-of-the-art composite kernel methods for *relation extraction*. All the training sets consist of all *relation candidates*, i.e., pairs of entities found in one sentence. Each example consists of a relation label, e.g. $merger$ or $nomerger$, the syntactic tree of the sentence in which the arguments occur, and several features which are now described in detail.

The *baseline-featureset* contains the word-features proposed by [Zhou *et al.*, 2005]. These features are mainly based on words of the *relation candidate* entities or words nearby in the sentence. For its use by the tree kernel, the feature set also contains the parse tree of the sentence the *relation candidate* is extracted from.

The *word-vector-featureset* contains just the word-vector of the sentence from which the *relation candidate* is extracted, and the parse tree.

The *big-word-vector-featureset* contains just the word-vector, the parse tree and the *baseline-featureset*.

The *stem-x-tree-featuresets* are equal to the *word-vector-featureset* but the parse tree contains the word stems inserted at depth-level x or as a replacement of the original terminal symbol.

The parse tree is given by running the Stanford parser [Klein and Manning, 2002] trained on the NEGRA corpus [Skut *et al.*, 1997]. We applied 10-fold cross validation using the composite kernel with a parameter setting of $C = 2.4$, $m = 3$ and $\alpha = 0.6$ (see Section 2.2).

Performance

Table 1 shows the performance of the state of the art method and the two versions of our new method. Table 2 shows the standard deviation of the performance measures in 10-fold cross validation.

Table 1: Performance of *relation extraction* on the companies corpus using 10-fold cross validation.

Featureset	Precision	Recall	F-meas.
baseline	33.47%	52.27%	38.64%
word-vector	36.41%	69.93%	45.45%
big-word-vector	36.83%	74.86%	**48.73%**
stem-replace-tree	31.46%	**76.03%**	44.08%
stem-0-tree	37.94%	47.90%	41.79%
stem-1-tree	**44.33%**	53.42%	47.51%
stem-2-tree	36.28%	62.91%	45.64%

As can be seen, recall increases significantly using word vectors in the linear kernel and word stems in the tree kernel while at the same time the deviation decreases. Precision is best when semantic information in the tree is used at level 1. The best F-measure achieved by the big-word-vector is to be explained by the very few sentences containing a

Table 2: Standard deviation of the performance of *relation extraction*

Precision	Recall	F-meas.	
baseline	**3.88%**	21.99%	8.88%
word-vector	12.16%	11.64%	5.12%
big-word-vector	5.15%	9.55%	**3.12%**
stem-replace-tree	4.63%	**8.99%**	4.32%
stem-0-tree	6.29%	14.55%	9.07%
stem-1-tree	7.58%	9.89%	4.40%
stem-2-tree	3.95%	10.89%	4.62%

positive and a negative candidate of a relation. If sentences include either a positive or a negative example of a relation, the *relation extraction* is downgraded to sentence classification, where word vectors are a well suited representation. Hence, for *relation extraction*, the enhanced trees remain important.

6 Conclusions and Future Work

We proposed an economic network that is built up extracting semi-structured websites containing financial stock information.

The network – consisting of entities and relations between them – should be kept up to date automatically. Therefore we presented an enhancement to state-of-the-art *relation extraction* methods. Our enhancements take into account the problems German language faces in contrast to the well-examined English language.

To evaluate our method we extracted a German document corpus of the economic domain. We tagged all the firms in our corpus and extracted all possible *relation candidates*. We tested state-of-the-art *relation extraction* methods on our *relation extraction* corpus and compared the results with the results achieved by our enhancements.

Our enhanced composite kernel method achieves significantly better performance compared to the baseline. Although using just the linear kernel performs best, the usage of the composite kernel will be needed if the relations become more frequent and the number of relation-types becomes bigger.

Future work will implement better measures for the evaluation, so that sentence classification effects in *relation extraction* can properly be detected. Our approach should be evaluated on English benchmark datasets. Additionally our approach to add semantic information in the parse trees could be replaced by using dependency trees. But unfortunately the used library (stanford parser) just offers trained dependency parsers for English and Chinese. Using dependency trees therefore might be tested on English datasets.

References

[Buncescu and Mooney, 2005] Razvan C. Buncescu and Raymond J. Mooney. A shortest path dependency kernel for relation extraction. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, British Columbia, Canada, 2005. Association for Computational Linguistics.

[Bunescu and Mooney, 2006] Razvan C. Bunescu and Raymond J. Mooney. Subsequence kernels for relation extraction. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 171–178. MIT Press, 2006.

[Collins and Duffy, 2001] Michael Collins and Nigel Duffy. Convolution kernels for natural language. In *Advances in Neural Information Processing Systems 14*, pages 625–632. MIT Press, 2001.

[Debnath *et al.*, 2008] Souvik Debnath, Niloy Ganguly, and Pabitra Mitra. Feature weighting in content based recommendation system using social network analysis. In *WWW 2008*. ACM Press, 2008.

[Domingos and Richardson, 2001] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Procs. KDD*, pages 57–66. ACM Press, 2001.

[Golder *et al.*, 2007] Scott A. Golder, Dennis M. Wilkinson, and Bernardo A. Huberman. Rhythms of social interaction: Messaging within a massive online network. In *Procs. 3rd Intl. Conf. on Communities and Technologies*, 2007.

[Haussler, 1999] David Haussler. Convolution kernels on discrete structures. Technical report, University of California in Santa Cruz, Computer Science Dept., 1999.

[Joachims, 1989] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Procs. Of European Conference on Machine Learning*, pages 137 – 142. Springer, 1989.

[Jungermann, 2009] Felix Jungermann. Information extraction with rapidminer. In Wolfgang Hoeppner, editor, *Proceedings of the GSCL Symposium 'Sprachtechnologie und eHumanities'*, pages 50–61. Universität Duisburg-Essen, Abteilung für Informatik und Angewandte Kognitionswissenschaft Fakultät für Ingenieurwissenschaften, 2009.

[Klein and Manning, 2002] Dan Klein and Christopher D. Manning. Fast extract inference with a factored model for natural language parsing. In *Proceedings of Advances in Neural Information Processing Systems*, 2002.

[Lin, 2004] Linguistic Data Consortium. *The ACE 2004 Evaluation Plan*, 2004.

[Mierswa *et al.*, 2006] Ingo Mierswa, Michael Wurst, Ralf Klinkenberg, Martin Scholz, and Timm Euler. Yale: Rapid prototyping for complex data mining tasks. In *Procs. 12th ACM SIGKDD Int. Conf. .Knowledge Discovery and Data Mining (KDD)*, 2006.

[Moschitti, 2006] Alessandro Moschitti. Efficient convolution kernels for dependency and constituent syntactic trees. In Johannes Fuernkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Procs. ECML*, pages 318 – 329. Springer, 2006.

[O'Madadhain *et al.*, 2003] Joshua O'Madadhain, Danyel Fisher, Scott White, and Yan-Biao Boey. The JUNG (java universal network/graph) framework. Technical Report Technical Report UCI-ICS 03-17, School of Information and Computer Science University of California, Irvine, CA 92697-3425, 2003.

[Palau *et al.*, 2004] Jordi Palau, Miquel Montaner, Beatriz López, and Josep Lluís De La Rosa. Collaboration analysis in recommender systems using social networks. In *Procs. Cooperative Information Agents VIII*, pages 137–151. Springer, 2004.

[Skut *et al.*, 1997] Wojciech Skut, Brigitte Krenn, Torsten Brants, and Hans Uszkoreit. An annotation scheme for free word order languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP)-97*, Washington, DC, 1997.

[Vishwanathan and Smola, 2002] S. V. N. Vishwanathan and Alexander J. Smola. Fast kernels for string and tree matching. In *NIPS*, pages 569–576, 2002.

[Zelenko *et al.*, 2003] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106, 2003.

[Zhang *et al.*, 2006] Min Zhang, Jie Zhang, Jian Su, and Guodong Zhou. A composite kernel to extract relations between entities with both flat and structured features. In *Procs. 44th Annual Meeting of ACL*, pages 825–832, 2006.

[Zhang *et al.*, 2007] Min Zhang, Wanxiang Che, Ai Ti Aw, Chew Lim Tan, Guodong Zhou, Ting Liu, and Sheng Li. A grammar-driven convolution tree kernel for semantic role classification. In *Procs. 4th Annual Meeting of ACL*, pages 200 – 207, 2007.

[Zhao and Grishman, 2005] Shubin Zhao and Ralph Grishman. Extracting relations with integrated information using kernel methods. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 419–426, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[Zhou *et al.*, 2005] Guodong Zhou, Jian Su, Min Zhang, and Jie Zhang. Exploring various knowledge in relation extraction. In *ACL*, pages 427 – 434, 2005.

[Zhou *et al.*, 2007a] Ding Zhou, Sergey A. Orshanskiy, Hongyuan Zha, and C. Lee Giles. Co-ranking authors and documents in a heterogeneous network. In *7th IEEE ICDM*, 2007.

[Zhou *et al.*, 2007b] GuoDong Zhou, Min Zhang, Dong Hong Ji, and QiaoMing Zhu. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2007.

Testing and Evaluating Tag Recommenders in a Live System

Robert Jäschke, Folke Eisterlehner, Andreas Hotho, and Gerd Stumme
Knowledge & Data Engineering Group
University of Kassel
Wilhelmshöher Allee 73
34121 Kassel, Germany
http://www.kde.cs.uni-kassel.de/

Abstract

The challenge to provide tag recommendations for collaborative tagging systems has attracted quite some attention of researchers lately. However, most research focused on evaluation and development of appropriate methods rather than tackling the practical challenges of how to integrate recommendation methods into real tagging systems, record and evaluate their performance.

In this paper we describe the tag recommendation framework we developed for our social bookmark and publication sharing system BibSonomy. With the intention to develop, test, and evaluate recommendation algorithms and supporting cooperation with researchers, we designed the framework to be easily extensible, open for a variety of methods, and usable independent from BibSonomy. Furthermore, this paper presents an evaluation of two exemplarily deployed recommendation methods, demonstrating the power of the framework.

1 Introduction

Collaborative tagging systems are web based systems that allow users to assign keywords – so called *tags* – to arbitrary resources. Tags are used for navigation, finding resources and serendipitous browsing and thus provide an immediate benefit for users. These systems usually include tag recommendation mechanisms easing the process of finding good tags for a resource. Delicious,[1] for instance, had a tag recommender in June 2005 at the latest,[2] BibSonomy[3] since 2006. Typically, such a recommender suggests tags to the user when she is annotating a resource. Recommending tags can serve various purposes, such as: increasing the chances of getting a resource annotated, reminding a user what a resource is about and consolidating the vocabulary across the users. Furthermore, as Sood et al. [Sood *et al.*, 2007] point out, tag recommendations "fundamentally change the tagging process from generation to recognition" which requires less cognitive effort and time.

Our contributions with this paper are: (i) presenting and evaluating a tag recommendation framework deployed in BibSonomy, an open collaborative tagging system, (ii) providing researchers a testbed to test and evaluate their methods in a live system, and (iii) showing first results which indicate the power of the framework to improve recommendation performance by clever selection strategies.

This paper is structured as follows: In Section 2 we introduce BibSonomy and motivate the task of tag recommendations; in Section 3 we review related work in the field and continue in Sec. 4 to explain the details of our tag recommendation framework. Then we elaborate on the evaluation methods (cf. Sec. 5) we have used to gather the results presented in Section 6. The paper closes with a conclusion and ideas for future work.

2 Application

In this section we briefly introduce *BibSonomy*, the collaborative tagging system used to deploy our framework, define what a *folksonomy* is and how we can express some of its properties, and describe the *tag recommendation* task.

2.1 BibSonomy

As foundation and testbed for our framework we use the social bookmark and publication sharing system *BibSonomy* [Hotho *et al.*, 2006a] which is run by us. BibSonomy started as a students project in spring 2005 and since then has evolved into a system with more than 1,500 active users. The goal was to implement a system for organizing BibTEX entries in a way similar to bookmarks in Delicious – which was at that time becoming more and more popular. After integrating bookmarks as a second type of resource into the system and upon the progress made, BibSonomy was opened for public access at the end of 2005 – first announced to colleagues only, later in 2006 to the public.

Users of BibSonomy can organize their bookmarks (URLs, favourites) and publication references by annotating them with tags. Plenty of features support them in their work: groups, tag editors, relations, various import and export options, etc. In particular, a REST-like [Fielding, 2000] API[4] eases programmatic interaction with BibSonomy and is the cornerstone of external cooperation with the presented tag recommendation framework. Technically, BibSonomy is based on several Java modules[5] which are merged in a Java Servlet/ServerPages based web application with an SQL database as backend.

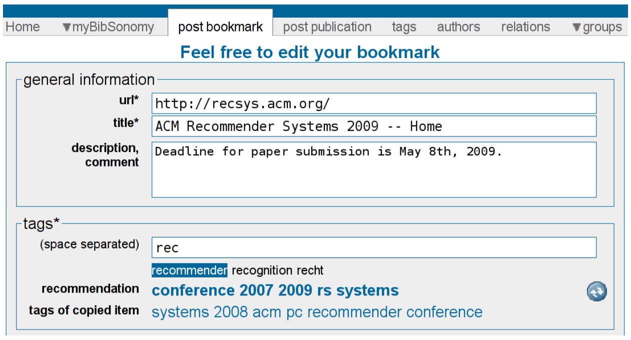
[1] http://delicious.com/

[2] http://www.socio-kybernetics.net/saurierduval/archive/2005_06_01_archive.html

[3] http://www.bibsonomy.org/

[4] http://www.bibsonomy.org/help/doc/api.html

[5] Some of them are freely available at http://dev.bibsonomy.org/.



Feel free to edit your bookmark

general information

url* http://recsys.acm.org/

title* ACM Recommender Systems 2009 -- Home

description, comment Deadline for paper submission is May 8th, 2009.

tags*

(space separated) rec

recommender recognition recht

recommendation **conference 2007 2009 rs systems**

tags of copied item systems 2008 acm pc recommender conference

Figure 1: BibSonomy's recommendation interface on the bookmark posting page. The 'tags' box contains a text input field where the user can enter the (space separated) tags, tags suggested for autocompletion, the tags from the recommender (bold), and the tags from the post the user just copies.

2.2 Folksonomy

A *folksonomy* is the datastructure underlying most collaborative tagging systems. It describes the assignment of *tags* by *users* to *resources*. Formally, a *folksonomy* is a tuple $\mathbb{F} := (U, T, R, Y)$ where U, T, and R are finite sets, whose elements are called *users*, *tags* and *resources*, resp., and Y is a ternary relation between them, i. e., $Y \subseteq U \times T \times R$, whose elements are called tag assignments (*tas* for short).[6]

Users are typically described by their user ID, and tags may be arbitrary strings. What is considered a resource depends on the type of system. For instance, in Delicious, the resources are URLs, in BibSonomy URLs or publication references, and in Last.fm, the resources are artists.

Building upon this model we can easily express certain properties of folksonomies, e. g., the number of tas of a given user u: $|Y \cap \{u\} \times T \times R|$, or the number of users which have tagged resource r with tag t: $|Y \cap U \times \{t\} \times \{r\}|$. To simplify matters, we define the set of all tags user u attached to resource r as $T_{ur} := \{t \in T \mid (u, t, r) \in Y\}$. Then a *post* is defined as (u, T_{ur}, r).

2.3 Tag Recommendations

Currently, tag recommendations in BibSonomy appear in two situations: when the user edits a bookmark or publication post. Since the part of the user interface showing recommendations is very similar for both the bookmark posting and the publication posting page, we show in Figure 1 the relevant part of the 'postBookmark'[7] page only.

Below the fields for entering URL, title, and a description (which are typically automatically filled), the 'tags' box keeps together the tagging information. There, the user can manually enter the tags to describe the resource. During typing the user is assisted by a JavaScript autocompletion which selects tags among the recommended tags and all of the user's previously used tags whose prefix matches the already entered letters. The suggested tags are shown directly below the tag input box (in the screenshot *recommender*, *recognition*, and *recht*). Further down there are in bold letters the five recommended tags ordered by their score from left to right. Thus, the recommender in action regarded *conference* to be the most appropriate tag for this resource and user. To the very right of the recommendation is a small icon depicting the *reload* button. It allows the user to request a new tag recommendation if he is unsatisfied with the one shown or wants to request further tags. We investigate the usage of this button in Sec. 6.2.

Besides triggering autocompletion with the tabulator key during typing, users can also click on tags with their mouse. They are then added to the input box. When the user copies a resource from another user's post, the tags the other user used to annotate the resource are shown below the recommended tags ('tags of copied item'). They are also regarded for autocompletion.

More formally, the tag recommendation task is: Given a resource r and a user u who wants to annotate r, the recommender shall return a set of recommended tags $T(u, r) := \{t_1, \ldots, t_k\}$ together with a *scoring function* $f \colon T(u, r) \to [0, 1]$ which assigns to each tag a score.[8] The value of k is fixed to 5 throughout this paper.

3 Related Work

Although having a different recommendation target (resources rather than tags), the REFEREE framework described by Cosley et al. [Cosley *et al.*, 2002] is most closely related to our work. It provided recommendations for the CiteSeer (formerly ResearchIndex) digital library. REFEREE recommends scientific articles to users of ResearchIndex while they search and browse. An open architecture allows researchers to integrate their methods into REFEREE. Besides the different recommendation target, the focus of the work is more on the evaluation of several different strategies than on the details of the framework.

A powerful, open, and well documented framework for recommendations is the Duine Framework[9] developed by Novay. It is based on work by van Setten [van Setten, 2005] and has a focus on explicit user ratings and non reoccuring items, e. g., like in a movie recommendation scenario where one does not recommend movies the user has already seen. This is in contrast to tag recommendations, where re-occuring tags are a crucial requirement of the system. Similar to what we present in Section 4.2 the framework implements various hybrid recommenders. They have been studied extensively – for a survey see [Burke, 2002].

Another recommendation framework is the AURA project's 'TasteKeeper' [Green and Alexander,] from Sun Microsystems. Despite having not been described in the literature, it has a strong focus on collaborative filtering algorithms.

The topic of tag recommendations in social bookmarking systems has attracted quite a lot of attention in the last years. Most related work describes recommendation approaches which could be used within our framework. The existent approaches usually lay in the collaborative filtering and information retrieval areas [Mishne, 2006; Byde *et al.*, 2007; Sood *et al.*, 2007]. Xu et al. [Xu *et al.*, 2006] identify properties of good tag recommendations like high coverage of multiple facets, high popularity, or least-effort and introduce a collaborative tag suggestion approach. A goodness measure for tags, derived

[6] In the original definition [Hotho *et al.*, 2006b], we introduced additionally a subtag/supertag relation, which we omit here.

[7] Logged in users can access this page at http://www.bibsonomy.org/postBookmark.

[8] Although, of course, f also depends on u and r, we will omit those two variables to simplify notation. Since f always appears together with $T(u, r)$, it should be clear from context, which f is meant.

[9] http://duineframework.org/

from collective user authorities, is iteratively adjusted by a reward-penalty algorithm. Further examples include Basile et al. [Basile *et al.*, 2007], suggesting an architecture of an intelligent tag recommender system, and Vojnovic et al. [Vojnovic *et al.*, 2007], trying to imitate the learning of the true popularity ranking of tags for a given resource during the assignment of tags by users.

Heymann et al. [Heymann *et al.*, 2008] model the tag prediction task as a binary classification problem for each tag with the web pages being the objects to classify. Besides the content of web pages, they also incorporate the anchor texts of links pointing to the page and host names of in-/outlinks as features for a support vector machine (SVM). They try to answer questions like “What precision can we get with low recall?”, “Which page information is best for predicting tags?”, or “What makes a tag predictable?”. Additionally, they apply association rules between tags to expand tag-based queries. Another analysis of the application of classification methods to the tag recommendation problem can be found in [Illig *et al.*, 2009 to appear].

One task of the 2008 ECML PKDD Discovery Challenge [Hotho *et al.*, 2008] also addressed the problem of tag recommendations in folksonomies. Tatu et al. [M. Tatu and D’Silva, 2008] base their suggestions on normalized tags from posts and normalized concepts from textual content of resources. This includes user added text like title or description as well as the document content. Using NLP tools they extract important concepts from the textual metadata and normalize them using Wordnet. Lipczak [Lipczak, 2008] developed a three step approach which utilizes words from the title expanded by a folksonomy driven lexicon, personalized by the tags of the posting user. Katakis et al. [I. Katakis and Vlahavas, 2008] consider the recommendation task as a multilabel text classification problem with tags as categories.

In [Jäschke *et al.*, 2008] we evaluated several tag recommendation methods on three large scale folksonomy datasets. The most successful algorithm, the graph-based FolkRank [Hotho *et al.*, 2006b], was followed by simpler approaches based on co-occurence counts and by collaborative filtering.

4 A Recommendation Framework for BibSonomy

Implementing a tag recommendation framework requires to tackle several challenges. For example, having enough data available for recommendation algorithms to produce helpful recommendations is an important requirement. The recommender needs access to the systems database and to what the user is currently posting (which could be accomplished, e.g., by (re)-loading recommendations using techniques like AJAX). Further data – like the full text of documents – could be supplied to tackle the cold-start problem (e. g., for content-based recommenders). Further aspects which should be taken into account include implementation of logging of user events (e. g., clicking, key presses, etc.) to allow for efficient evaluation of the used recommendation methods in an online setting. Together with a live evaluation this also allows us to tune the result selection strategies to dynamically choose the (currently) best recommendation algorithm for the user or resource at hand. The multiplexing of several available algorithms together with the simple inclusion of external recommendation services (by providing an open recommendation interface) is one of the benefits of the proposed framework.

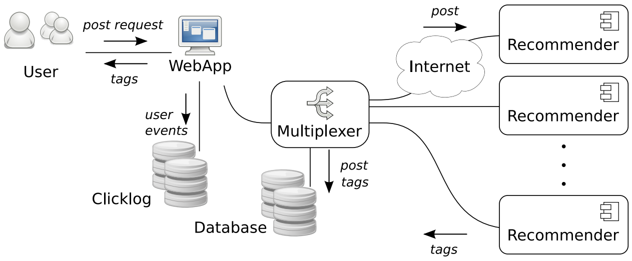


Figure 2: A schematic posting process.

Figure 2 gives an overview on the components of BibSonomy involved in a recommendation process. The web application receives the user’s HTTP request and queries the multiplexer (cf. Sec. 4.4) for a recommendation – providing it post information like URL, title, user name, etc.. Besides, click events are logged in a database (see Sec. 5.3). The multiplexer then requests the active recommenders to produce recommendations and selects one of the results. The suggested tags and the post are then logged in a database and the selected recommendation returned to the user.

4.1 Recommender Interface

One central element of the framework is the recommender interface. It specifies which data is passed from a recommendation request to one of the implemented recommenders and how they shall return their result. Figure 3 shows the UML class diagram of the *TagRecommender* interface one must implement to deliver recommendations to BibSonomy.

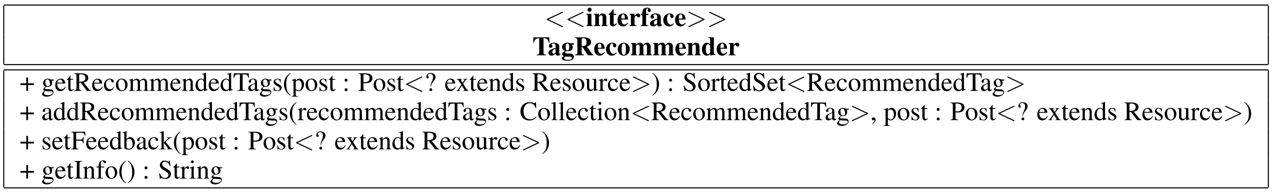
We decided to keep the interface as simple as possible by requiring only three methods, building on BibSonomy’s existing data model (Post, Tag, etc.) and adding as few classes as possible (RecommendedTag, RecommendedTagComparator).

The *getRecommendedTags* method returns – given a post – a sorted set of tags; *addRecommendedTags* adds to a given (not necessarily empty) collection of tags further tags. Since – given a post and an empty collection – *addRecommendedTags* should return the same result as *getRecommendedTags*, the latter can be implemented by delegation to the former. Nonetheless, we decided to require both methods to cover the simple ‘give me some tags’ case as well as more sophisticated usage scenarios (think of ‘intelligent’ collection implementations, or a recommender which improves given recommendations).

The post given to both methods contains data like URL, title, description, date, user name, etc. that will later be stored in the database and that the recommender can use to produce good recommendations. It might also contain tags, i. e., when the user edits an existing post or when he has already entered some tags and requests new recommendations. Implementations could use those tags to suggest different tags or to improve their recommendation.

With the *setFeedback* method the final post as it is stored in the database is given to the recommender such that it can measure and potentially improve its performance. Additionally, the *postID* introduced in Section 5.3 is contained in the post (as well as in the post of the first two methods) such that the recommender can connect the post with the recommended tags it provided.

Finally, the *getInfo* method allows the programmer to



<<**interface**>>
TagRecommender

+ getRecommendedTags(post : Post<? extends Resource>) : SortedSet<RecommendedTag>
+ addRecommendedTags(recommendedTags : Collection<RecommendedTag>, post : Post<? extends Resource>)
+ setFeedback(post : Post<? extends Resource>)
+ getInfo() : String

Figure 3: The UML class diagram of the tag recommender interface.

provide some information describing the recommender. This can be used to better identify recommenders or be shown to the user.

Two further classes augment the interface: The *RecommendedTag* class basically extends the *Tag* class as used in the BibSonomy API (cf. Sec. 2.1) by adding floating point *score* and *confidence* attributes. A corresponding *RecommendedTagComparator* can be used to compare tags, e. g., for sorted sets. It first checks textual equality of tags (ignoring case) and then sorts them by score and confidence. Consequently, tags with equal names are regarded as equal.

Our implementation is based on Java and all described classes are contained in the module *bibsonomy-model*, which is available online as JAR file in a Maven2 repository.[10] However, implementations are not restricted to Java – using the remote recommender (see Sec. 4.3) one can implement a recommender in any language which is then integrated using XML over HTTP requests.

4.2 Meta Recommender

Meta or *hybrid recommenders* [Burke, 2002] do not generate recommendations on their own but instead call other recommenders and modify or merge their results. Since they implement the same interface, they can be used like any other recommender. More formally, given n recommendations $T_1(u,r), \dots, T_n(u,r)$ and corresponding scoring functions $f_1, \dots, f_n$, a meta recommender produces a merged recommendation $T(u,r)$ with scoring function f. The underlying design pattern known from software architecture is that of a *Composite*.

As we will see in Section 4.5, meta recommenders allow the building of complex recommenders from simpler ones and thus simplify implementation and testing of algorithms and even stimulate development of new methods. Furthermore, they allow for flexible configuration, since their underlying recommenders can be exchanged at runtime. This section introduces the meta recommenders that are currently used in our framework.

First Weighted By Second

As an example of a cascade hybrid, the idea behind this recommender is to re-order the tags of one recommendation using scores from another recommendation. More precisely, given recommendations $T_1(u,r)$ and $T_2(u,r)$ and corresponding scoring functions f_1 and f_2, this recommender returns a recommendation $T(u,r)$ with scoring function f, which contains all tags from T_1 which appear in T_2 (with $f(t) := f_2(t)$) plus all the remaining tags from T_1 (with lower f but respecting the order induced by f_1). If $T_1(u,r)$ does not contain enough recommendations, T is filled by the not yet used tags from $T_2(u,r)$ – again with f being lower than for the already contained tags and respecting the order induced by f_2.

Weighted Merging

This weighted hybrid recommender enables merging of recommendations from different sources and weighting of their scores. Given n recommendations $T_1(u,r), \dots, T_n(u,r)$, corresponding scoring functions $f_1, \dots, f_n$, and (typically fixed) weights $\rho_1, \dots, \rho_n$ (with $\sum_{i=1}^{n} \rho_i = 1$), the weighted merging recommender returns a recommendation $T(u,r) := \bigcup_{i=1}^{n} T_i(u,r)$ and a scoring function $f(t) := \sum_{i=1}^{n} \rho_i f_i(t)$ (with $f_i(t) := 0$ for $t \notin T_i(u,r)$).

4.3 Remote Recommender

The remote recommender retrieves recommendations from an arbitrary external service using HTTP requests in REST-based [Fielding, 2000] interaction. Therefore, it uses the XML schema of the BibSonomy REST-API.[11] This recommender has three advantages: it allows us to distribute the recommendation work over several machines, it opens the framework to include recommenders from auxilliary partners, and it enables programming language independent interaction with the framework.

To simplify implementation of external recommenders, we provide an example web application needing almost zero configuration to include a custom Java recommender.[12] Furthermore, we plan to integrate recommendations into BibSonomy's API to allow clients retrieve recommendations (e. g., such that the Firefox browser add-on can show recommendations during bookmark posting).

4.4 Multiplexing Tag Recommender

Our framework's technical core component is the so called *multiplexing tag recommender* (see Fig. 2). Implementing BibSonomy's tag recommender interface, it provides the web application with tag recommendations, using one of the recommenders available. All recommendation requests and each recommender's corresponding result are logged in a database (see Sec. 5.3). For this purpose, every tag recommender is registered during startup and assigned to a unique identifier. For technical reasons, we differentiate between locally installed and remote recommenders (cf. Sec. 4.3).

Whenever the *getRecommendedTags* method is invoked, the corresponding recommendation request is delegated to each recommender, spawning separate threads for each recommender. After a timeout period of 100 ms, one of the collected recommendations is selected, applying a preconfigured *selection strategy*:

For our evaluation procedure we implemented a '*sampling without replacement*' strategy which randomly chooses exactly one recommender and returns all of its recommended tags. If the user requests recommendations

[10] `http://dev.bibsonomy.org/maven2/org/bibsonomy/bibsonomy-model/`

[11] `http://www.bibsonomy.org/help/doc/xmlschema.html`

[12] `http://dev.bibsonomy.org/maven2/org/bibsonomy/bibsonomy-recommender-servlet`

more than once during the same posting process (e. g., by using the 'reload' button), the strategy selects recommendations from a recommender the user has not seen during this process.

4.5 Example Recommender Implementations

Using the proposed framework, we implemented several recommendation methods, whereas two of them are currently active in BibSonomy. Both build upon the meta recommenders described in Section 4.2 and simpler recommenders which we describe only briefly because they are fairly self-explanatory. The short names in parentheses are for later reference.

Most Popular ρ-Mix (MPρ-mix)

Motivated by the good results of mixing tags which often have been attached to the resource with tags the user has often used, we implemented a variant of the *most popular ρ-mix* recommender described in [Jäschke *et al.*, 2008]. The recommender has been implemented as a combination of three recommenders, using a value of $\rho = 0.6$:

1. the *most popular tags by resource* recommender which returns the k tags $T_1(u,r)$ which have been attached to the resource most often (with $f_1(t) := \frac{|Y \cap U \times \{t\} \times \{r\}|}{|Y \cap U \times T \times \{r\}|}$, i. e., the relative tag frequency),
2. the *most popular tags by user* recommender which returns the k tags $T_2(u,r)$ the user has used most often (with $f_2(t) := \frac{|Y \cap \{u\} \times \{t\} \times R|}{|Y \cap \{u\} \times T \times R|}$, i. e., the relative tag frequency), and
3. the *weighted merging* meta recommender described in Section 4.2 which merges the tags of the two former recommenders, with weights $\rho_1 = \rho = 0.6$ and $\rho_2 = 1 - \rho = 0.4$.

Title Tags Weighted by User Tags (TbyU)

Inspired by the first recommender implemented in BibSonomy [Illig, 2006] and by similar ideas in [Lipczak, 2008], we implemented a recommender which ranks tags extracted from the resource's title using the frequency of the tags used by the user. Technically, this is again a combination of three recommenders:

1. a simple *content based recommender*, which extracts k tags $T_1(u,r)$ from the title of a resource, cleans them and checks against a multilingual stopword list,
2. the *most popular tags by user* recommender as described in the previous section – here returning *all* tags $T_2(u,r)$ the user has used (by setting $k = \infty$), and
3. the *first weighted by second* meta recommender described in Section 4.2 which weights the tags from the content based recommender by the frequency of their usage by the user as given by the second recommender.

Other

Besides the simple recommenders introduced along the MPρ-mix and TbyU recommender, we have implemented recommenders for testing purposes (a *fixed tags recommender* and a *random tags recommender*), a recommender which proposes tags from a web page's HTML meta information keywords, as well as a recommender using the FolkRank algorithm [Hotho *et al.*, 2006b].

More complex recommenders can be thought of, e. g., a nested *first weighted by second* recommender, whose first recommender is a *weighted merging* meta recommender merging the suggestions from a *content based recommender* and a *most popular tags by resource* recommender and then scoring the tags by the scores from the *most popular tags by user* recommender.

Listing 1: The Java method used to clean tags.

```
public String cleanTag(String tag) {
  return Normalizer.normalize(tag,
    Normalizer.Form.NFKC).
      replaceAll(”[ˆ0−9\\p{L}]+”, ””).
      toLowerCase();
}
```

5 Evaluation

We evaluate the performance of a recommender by comparing the tags it suggested with the tags used to annotate a resource. Then recall ('Which fraction of the used tags could be suggested?') and precision ('Which fraction of the suggested tags was used?') quantify the quality of the recommendation. Furthermore, the logging of click events allows us to evaluate the user behavior in more detail.

5.1 Measures

As performance measures we use precision, recall, and f1-measure (f1m) which are standard in such scenarios [Herlocker *et al.*, 2004]. For each post (u, T_{ur}, r) we compare the recommended tags $T(u,r)$ with the tags T_{ur} the user has finally assigned. Then, precision and recall of a recommendation are defined as follows

$$\text{recall}(T(u,r)) = \frac{|T_{ur} \cap T(u,r)|}{|T_{ur}|} \tag{1}$$

$$\text{precision}(T(u,r)) = \frac{|T_{ur} \cap T(u,r)|}{|T(u,r)|} \quad . \tag{2}$$

We then average these values over all posts in the given set and compute the f1-measure as

$$\text{f1m} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad .$$

5.2 Data Cleansing

Before intersecting T_{ur} with $T(u,r)$, we clean the tags in both sets according to the Java method *cleanTag* shown in Listing 1. This means, we ignore the case of tags and remove all characters which are neither numbers nor letters.[13] Since we assume all characters to be UTF-8 encoded, the method will *not* remove umlauts and other non-latin characters. We also employ unicode normalization to normal form KC.[14] Finally, we ignore tags which are 'empty' after normalization (i. e., they neither contained a letter nor number) or which are equal to the strings *imported*, *public*, *systemimported*, *nn*, *systemunfiled*. Thus, in the following we always regard cleaned tags.

[13] See also the documentation of `java.util.regex.Pattern` at `http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html`.

[14] `http://www.unicode.org/unicode/reports/tr15/tr15-23.html`

5.3 Logging

For evaluating performance of the tag recommenders available, we store in a database for each recommendation process the corresponding bookmark or BibTeX entry as well as each recommender's recommendation, identified by a unique *recommendationID*. Furthermore, the applied selection strategy together with the recommenders and tags selected are stored.

Several recommendation requests may refer to a single posting process (e. g., when the user pressed the 'reload' button). For identifying these correspondences, a random identifier (*postID*) is generated whenever a post or editing process is started and retains valid until the corresponding post is finally stored in BibSonomy. This *postID* is mapped to each corresponding *recommendationID*. At storage time, the *postID* together with the corresponding user name, time stamp and a hash identifying the resource is stored. This connects each post of each user with all referring recommendations and vice versa.

Additionally, the user interaction is tracked by logging mouse click events using JavaScript. Each click on one of BibSonomy's web pages is logged using AJAX into a separate logging table. Information like the shown page, the DOM path of the clicked element, the underlying text, etc is stored.[15]

6 Results

The following analysis is based on data from posting processes between May 15th and June 26th 2009; this is ongoing work – this analysis is the first step of a long term study of the BibSonomy recommendation framework. Only public posts from users not flagged as spammer were taken into account. Since tag recommendations are provided in the web application only when *one* resource is posted, posts originating from automatic import (e. g., Firefox bookmarks, or BibTeX files) or BibSonomy's API are not contained in the analysis.

6.1 General

We start with some general numbers: In the analysed period, 5,840 posting processes (3,474 for BibTeX, 2,366 for bookmarks) have been provided with tag recommendations. The MPρ-mix recommender served recommendations for 2,935 postings, the TbyU recommender for 3,006. Their precision and recall is depicted in Figure 4. On the plotted curve, from left to right the number of evaluated tags increases from one to five. I. e., we first regard only the tag t with the highest value $f(t)$, then the two tags with highest f, and so on. Thus, the more recommended tags are regarded, recall increases while precision decreases. In general, both precision and recall are rather low with the MPρ-mix recommender performing better than the TbyU recommender.

6.2 Influence of the 'reload' Button

Since users can request to reload recommendations when posting a resource, we here investigate the influence of the 'reload' button. Is the first recommendation sufficient or do users request another recommendation? Are recommendations which got replaced by the user pressing the 'reload' button worse than those shown last? Has one recommender more often been reloaded than the other?

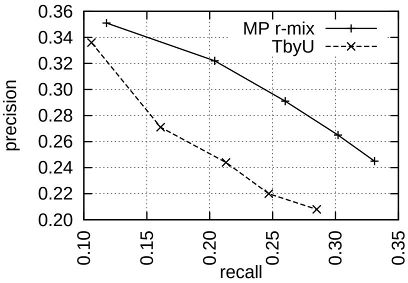


Figure 4: Precision and Recall

Table 1: The influence of the 'reload' button.

measure	#posts		f1m@5	
recommender $\mathfrak{r}$	MPρ-mix	TbyU	MPρ-mix	TbyU
$F_\mathfrak{r} \setminus L_\mathfrak{r}$	337	319	0.258	0.270
$L_\mathfrak{r} \setminus F_\mathfrak{r}$	331	363	0.380	0.364
$F_\mathfrak{r} \cap L_\mathfrak{r}$	2,271	2,339	0.277	0.224

In 767 (274 bookmark, 493 BibTeX) of the 5,840 posting processes the users requested to reload the recommendation. Thus, in around 13 % of all posting processes users requested another recommendation.

Several recommenders can be involved in one posting process. There is the recommendation which appears directly after loading the posting page (*first*), there are recommendations which appear after the user has pressed the 'reload' button, and there is the recommendation shown before the user finally saves the post (*last*). Thus, given a recommender $\mathfrak{r}$, we can define the set $F_\mathfrak{r}$ to contain those posts, where the recommender $\mathfrak{r}$ showed the first tags, and $L_\mathfrak{r}$ as the set of posts where recommender $\mathfrak{r}$ showed the last tags (i. e., before the post is stored).

For each recommender $\mathfrak{r}$ we can then look at the sets $F_\mathfrak{r} \setminus L_\mathfrak{r}$, $L_\mathfrak{r} \setminus F_\mathfrak{r}$, and $F_\mathfrak{r} \cap L_\mathfrak{r}$. Posts where the user did not press the reload button are contained in both $F_\mathfrak{r}$ and $L_\mathfrak{r}$ and thus in $F_\mathfrak{r} \cap L_\mathfrak{r}$. Table 1 shows the result of our analysis.

For both of the two deployed recommenders and for all three sets, the table shows the number of posts in the corresponding set, and the average f1m at the fifth tag.[16] As one can see, the number of posts where the reload button has not been pressed ($F_\mathfrak{r} \cap L_\mathfrak{r}$) is quite large for both recommenders (around $2,300$). There is also only little difference in the number of posts for the recommenders over the different sets, except the higher number of posts for the TbyU recommender in $L_\mathfrak{r} \setminus F_\mathfrak{r}$. It contains those posts, where the user requested to reload the recommendation and where the recommender at hand delivered the last recommendation. Thus, the TbyU recommender more often provided the last recommendation than the MPρ-mix recommender.

The most noticeable observation is the good performance of both recommenders for this set. Both precision and recall are much higher than for the other two sets. This suggests that the first suggestion was rather bad and caused the user to request another recommendation which indeed better fitted his needs. The worse values for $F_\mathfrak{r} \setminus L_\mathfrak{r}$ also support this thesis. A noteworthy difference between the two recommenders is the performance of the TbyU recommender for $F_\mathfrak{r} \setminus L_\mathfrak{r}$ which is better than its overall per-

[15] Note that users can disable logging on the settings page, thus not all posting processes yield clicklog events.

[16] We omit precision and recall, since whenever the f1m for one set was better/worse than for another set, precision and recall were better/worse, too.

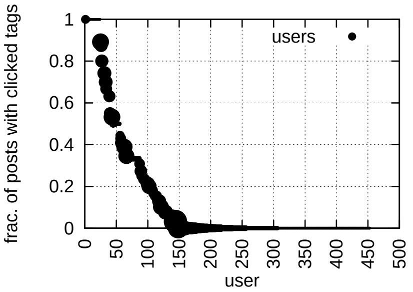


Figure 5: Users sorted by their fraction of click/noclick-posts; The y-axis depicts the fraction of posts where recomended tags were clicked.

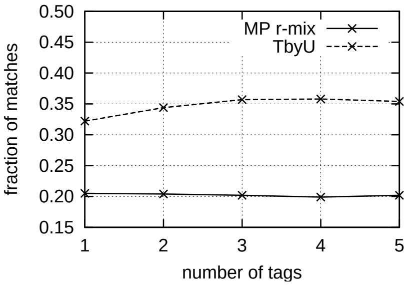


Figure 6: The fraction of matching tags which have been clicked.

formance (i. e., on $F_\mathfrak{r} \cap L_\mathfrak{r}$). This could be an indicator that those users which actively used the recommender (by pressing the 'reload' button) took better notice of this recommender's tag suggestions.

The usage of the 'reload' button is a good indicator for the interest of the user in the recommendations. However, the data we gathered during the evaluation period is still rather sparse and thus no final conclusions can be drawn.

6.3 Logged 'click' Events

Next we evaluate data from the log which records when a user clicked a recommended tag (cf. Sec. 5.3). Clicks are rather sparse: in only 1,061 (485 bookmark, 576 BIBTEX) of the 5,840 posting processes users clicked on a tag.

First, we want to answer the questions "How is clicking distributed over users?" and "Are there users which always/never click?". Figure 5 shows users sorted by the fraction of posting processes at which they have clicked on a recommended tag. The size of each circle depicts the logarithm of the user's number of posts. Closer to the left are users which in almost all posting events clicked on a recommendation; users closer to the right never clicked a tag during recommendation. Although only around 150 users clicked on a recommendation, half of the remaining users are represented by only one post. This could mean that only after some time users discover and use the recommendations. However, there are also some active users which almost never clicked on a recommendation.

In Figure 6 we see for each number of recommended tags (from one to five), the fraction of matches which stem from a click on the tag (instead of manual typing). For the TbyU recommender around 35 % of the matches come from the user clicking on a tag. Thus, although users infrequently click on tags, a large fraction of the correctly recommended tags of that recommender has been clicked instead of typed. Why there is a difference of around 15 % between the two recommenders with a higher click fraction for the TbyU recommender (in contrast to its worse f1m) is not clear. One explanation could be the different sources of tags the two recommenders use: while the MPρ-mix recommender delivers popular tags the user might have used before and thus can easily type, the TbyU recommender also suggests new and probably complicated tags extracted from the title which are easier to click than to type.

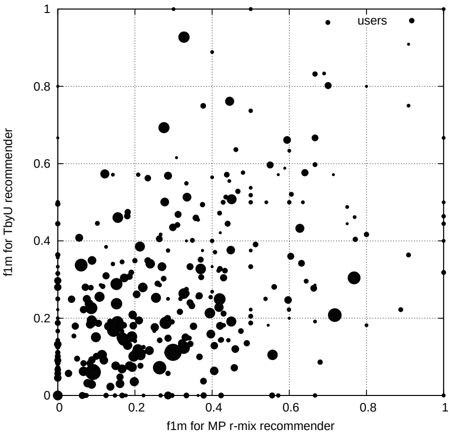


Figure 7: Average f1-measure for each user and recommender

6.4 Average F1-Measure per User

Which properties of a posting process could help a multiplexer strategy to smartly choose a certain recommender instead of randomly selecting one? For space reasons we focus on the user only – other characteristics could be likewise interesting (e. g., resource type or the recommended tags). Figure 7 shows the average f1m of the MPρ-mix recommender versus the average f1m of the TbyU recommender for each of the 380 users[17] in the data. In the plot, each user is represented by a circle whose size depicts the logarithm of the user's number of posts.

The most interesting users are reflected by the circles farthest from the diagonal, i. e., those users who have a high f1m for one but a low f1m for the other recommender. As one can see, such users exist even at higher post counts. Once such a user is identified, one could primarily select recommendations from the user's preferred recommender.

7 Conclusions and Future Work

In this paper, we presented the tag recommendation framework we developed for BibSonomy. It allows us to not only integrate and judge recommendations from various sources but also to develop clever selection strategies. A strength of the framework is its ability to log all steps of the recommendation process and thereby making it traceable. E. g., the diagrams and tables presented in this paper are automatically generated and will be integrated in a web application for analysing and controlling the framework and its recommenders.

As the results show, there is no clear picture which of the two recommendation methods performs better. There is a dependency on the number of regarded tags, the user at hand, and also slightly on the moment of recommendation. This suggests that we can achieve better performance

[17] Only users which got recommendations from *both* recommenders were taken into account.

not only by adding improved recommendation methods but also by implementing adaptive selection strategies. In case of the user dependency, one could prefer the better performing recommender by increasing its selection probability or even couple the probability with the current recommendation quality.

Finally, the framework was the cornerstone of this year's ECML PKDD Discovery Challenge,[18] where one task required the participants to deliver live recommendations for BibSonomy. This also was a larger stress test for external recommenders and the framework itself which it bravely passed. After that we opened the framework for interested researchers which we would like to encourage to contact us via an e-mail to webmaster@bibsonomy.org.

Acknowledgement Part of this research was funded by the European Union in the Tagora (FET-IST-034721) project and by the DFG in the project "Info 2.0 – Informationelle Selbstbestimmung im Web 2.0".

References

[Basile *et al.*, 2007] Pierpaolo Basile, Domenico Gendarmi, Filippo Lanubile, and Giovanni Semeraro. Recommending smart tags in a social bookmarking system. In *Bridging the Gap between Semantic Web and Web 2.0 (SemNet 2007)*, pages 22–29, 2007.

[Burke, 2002] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, November 2002.

[Byde *et al.*, 2007] Andrew Byde, Hui Wan, and Steve Cayzer. Personalized tag recommendations via tagging and content-based similarity metrics. In *Proc. of the Int. Conf. on Weblogs and Social Media*, March 2007.

[Cosley *et al.*, 2002] Dan Cosley, Steve Lawrence, and David M. Pennock. REFEREE: an open framework for practical testing of recommender systems using ResearchIndex. In *VLDB '02: Proceedings of the 28th international conference on Very Large Data Bases*, pages 35–46. VLDB Endowment, 2002.

[Fielding, 2000] Roy T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.

[Green and Alexander,] Steve Green and Jeff Alexander. The advanced universal recommendation architecture (AURA) project. `http://www.tastekeeper.com/`.

[Herlocker *et al.*, 2004] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[Heymann *et al.*, 2008] Paul Heymann, Daniel Ramage, and Hector Garcia-Molina. Social tag prediction. In *SIGIR '08: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 531–538, New York, NY, USA, 2008. ACM.

[Hotho *et al.*, 2006a] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. BibSonomy: A social bookmark and publication sharing system. In Aldo de Moor, Simon Polovina, and Harry Delugach, editors, *Proc. of the Conceptual Structures Tool Interoperability Workshop at the 14th Int. Conf. on Conceptual Structures*, Aalborg, Denmark, July 2006. Aalborg University Press.

[Hotho *et al.*, 2006b] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *Lecture Notes in Computer Science*, pages 411–426, Heidelberg, June 2006. Springer.

[Hotho *et al.*, 2008] Andreas Hotho, Beate Krause, Dominik Benz, and Robert Jäschke, editors. *ECML PKDD Discovery Challenge 2008 (RSDC'08)*, 2008.

[I. Katakis and Vlahavas, 2008] G. Tsoumakas I. Katakis and I. Vlahavas. Multilabel text classification for automated tag suggestion. In Hotho et al. [2008], pages 75–83.

[Illig *et al.*, 2009 to appear] Jens Illig, Andreas Hotho, Robert Jäschke, and Gerd Stumme. A comparison of content-based tag recommendations in folksonomy systems. In *Postproceedings of the International Conference on Knowledge Processing in Practice (KPP 2007)*, 2009 (to appear).

[Illig, 2006] Jens Illig. Entwurf und Integration eines Item-Based Collaborative Filtering Tag Recommender Systems in das BibSonomy-Projekt. Project report, 2006.

[Jäschke *et al.*, 2008] Robert Jäschke, Leandro Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in social bookmarking systems. *AI Communications*, 21(4):231–247, 2008.

[Lipczak, 2008] M. Lipczak. Tag recommendation for folksonomies oriented towards individual users. In Hotho et al. [2008], pages 84–95.

[M. Tatu and D'Silva, 2008] M. Srikanth M. Tatu and T. D'Silva. RSDC'08: Tag recommendations using bookmark content. In Hotho et al. [2008], pages 96–107.

[Mishne, 2006] Gilad Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *WWW '06: Proceedings of the 15th International Conference on World Wide Web*, pages 953–954, New York, NY, USA, 2006. ACM Press. paper presented at the poster track.

[Sood *et al.*, 2007] Sanjay Sood, Sara Owsley, Kristian Hammond, and Larry Birnbaum. TagAssist: Automatic tag suggestion for blog posts. In *Proc. of the Int. Conf. on Weblogs and Social Media (ICWSM 2007)*, 2007.

[van Setten, 2005] Mark van Setten. *Supporting people in finding information : hybrid recommender systems and goal-based structuring*. PhD thesis, University of Twente, Enschede, The Netherlands, December 2005.

[Vojnovic *et al.*, 2007] M. Vojnovic, J. Cruise, D. Gunawardena, and P. Marbach. Ranking and suggesting tags in collaborative tagging applications. Technical Report MSR-TR-2007-06, Microsoft Research, 2007.

[Xu *et al.*, 2006] Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the semantic web: Collaborative tag suggestions. In *Proc. of the Collaborative Web Tagging Workshop at the WWW 2006*, Edinburgh, Scotland, May 2006.

[18] `http://www.kde.cs.uni-kassel.de/ws/dc09`

Towards Cross-Community Effects in Scientific Communities*

[work in progress]

Marcel Karnstedt and **Conor Hayes**
Digital Enterprise Research Institute (DERI)
National University of Ireland, Galway, Ireland
first.last@deri.org

Abstract

Community effects on the behaviour of individuals, the community itself and other communities can be observed in a wide range of applications. This is true in scientific research, where communities of researchers have increasingly to justify their impact and progress to funding agencies. Previous work has tried to explain these phenomena by analysing co-citation graphs with methods from social network analysis and graph mining. More recent approaches have supplemented this with techniques from textual clustering. However, there is still a great potential for increasing the quality and accuracy of this analysis, especially in the context of cross-community effects. In this work, we present existing approaches and discuss their strengths and weaknesses. Based on this, we choose two closely related communities and propose novel ideas to detect and explain cross-community effects with a special focus on their characteristics in a given timeline. The outcome is a roadmap for advanced analysis of cross-community effects, which promises valuable insights for all areas of scientific research.

1 Introduction

Community structures can be found in a wide variety of applications. Analysing these structures provided very interesting insights into the internals and functioning of social networks since first works in this field appeared. This began with Milgram's famous experiment on the "six degrees of separation" [Milgram, 1967] and can be found in the whole research area on social network analysis [Granovetter, 1973].

In current days, this topic experiences amplified attraction again. This is due to the wide variety and great potential of social applications in the Web, such as Facebook[1] and Wikipedia[2]. Researchers as well as economists expect valuable outcomes for optimising Web technologies and increasing revenue from the detailed analysis of community structures and their effects. An evidence for the popularity of methods from social sciences is the existence of the "six degrees of Kevin Bacon"[3], in relation to Milgram's original work. A main problem of applying these techniques is their usually restricted scalability. Network structures found in the current Web tend to be by far too large for directly applying these methods, which are developed for small graphs around single individuals.

One specific field in this area, which is especially interesting for computer researchers, is the analysis of scientific communities. The practice of citing other authors' works is particularly typical for computer scientists. The detection and explanation of community effects helps to justify progress and gather funding as well as to identify trends and evolving fields over time. It can guide funding agencies and tenure committees to make more informed decisions. It has been shown that citation analysis is a very promising approach to detect correlations and interdependencies between different researchers and fields of research.

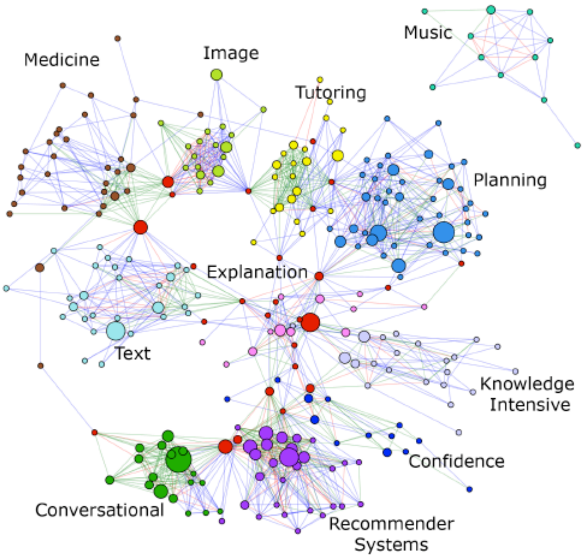


Figure 1: A co-citation graph for social network analysis [Greene *et al.*, 2009]

So far, most of these works focused on specific communities and the different sub-communities, i.e., different fields in one general area of computer research. But, the analysis of effects between different broader communities promises to reveal more and different insights. This can help to leverage inter-community communication and collaboration as well as to increase the impact of research. To approach this novel view on community analysis, we choose two closely related communities as a starting point. Our choice falls on the Semantic Web community, as a rather young but dramatically evolving one, and the Information Retrieval community, as a profiled and closely related community. Later, we plan to extend the analysis

*This material is based upon works jointly supported by the Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2) and under Grant No. 08/SRC/I1407 (Clique: Graph & Network Analysis Cluster)

[1] http://www.facebook.com

[2] http://www.wikipedia.org

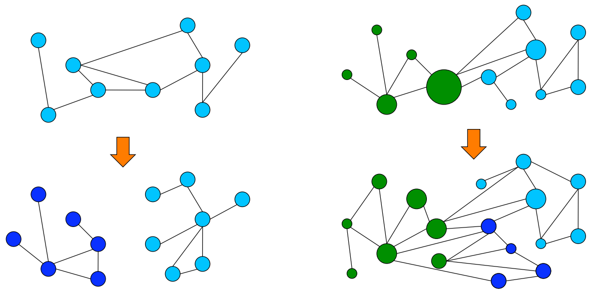
[3] http://www.thekevinbacongame.com

to other related communities, such as that from Database research. The goal is to develop techniques for analysing cross-community effects for an arbitrary number of communities. We expect valuable insights not only for the chosen fields of research, but also for scientific research in general. The outcomes of this work will have great potential for understanding, directing and optimising the effects and interdependencies between the identified groups of scientific research.

The general approach starts with collecting a set of seed papers. For these works, citations have to be extracted. Based on this, a graph of citation relations can be build, where the vertices represent works or authors and the edges between nodes represent relations between them based on citations. Figure 1 shows an example of such a graph, taken from [Greene *et al.*, 2009]. The graph shows the community structure in the field of Case-Based Reasoning (CBR) in the state of the year 2008. The figure shows the structure of one community, but also indicates how the structure between communities could look like. On top of such a graph, methods from social network analysis and graph mining will help to understand the specific cross-community effects. In Section 2, we give brief insights into the phenomena we expect. Finally, we provide a roadmap for following works, whereby we focus on the three main aspects of data gathering (Section 3.1), graph construction (Section 3.2) and actual analysis (Section 3.3). Section 4 concludes the paper.

2 Expected Phenomena

There is a wide range of phenomena that we expect to find with the methods proposed in this work. In this section, we briefly describe two selected ones, namely the paradigm shift [Kuhn, 1996] and paradigm merge. These are two effects that play a specific role especially for the analysis of scientific communities and the cross-community relations between them.



(a) Paradigm shift

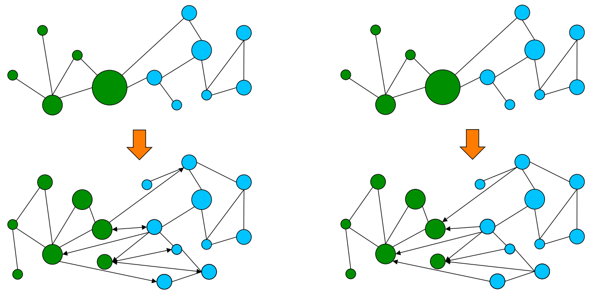
(b) Paradigm merge (with paradigm shift)

Figure 2: Paradigm shift and paradigm merge as possible phenomena

Figure 2(a) illustrates what can be called a paradigm shift in scientific communities. The upper part shows citation relations between different authors or works that might be found at a specific point in time. Analysing the development of this graph over time might reveal that a sub-community somehow detaches from its original community. This means, authors from both communities do not cite each other any more, with ongoing time the sub-community seems to “speak a different language” that is not understood by the remaining community any more. Such a phenomenon was first described by Kuhn and called a paradigm shift [Kuhn, 1996]. Clearly, to detect such an effect, the citation structure has to be analysed over time, by looking at the corresponding graphs from different points in time.

In Figure 2(b) we show the opposite of this, which we call a paradigm merge. Such an effect can be expected particularly when analysing two or more originally separated communities. Over time, the communities approach each other, represented by more and more citings between them. This can lead to closely related communities or even to a merge into one larger community. For some communities, we even expect a combination of paradigm shift and merge. This means, from one large community only a sub-community approaches the originally separated one – whereby in parallel detaching from its original field of research. We indicate this in the figure by the different shades of the nodes.

Especially for new and rapidly evolving communities like the field of Semantic Web research we expect this to be observed in combination with another effect. In its beginning, the only work that is “visible” to other communities might be a very fundamental and ground-breaking contribution by one of its founders. For the Semantic Web, one can think of Tim Berners-Lee famous work [Berners-Lee *et al.*, 2001], which is seen as the initial work founding that community. We indicate such a visibility by using differently sized nodes in the figure. Over time, more works “appear on the horizon”, the coastline of the community becomes visible and more islands are cited. This results in a shift of visibility between the actors of the evolving community. Analysing such effects can be done by looking at the citation graph accumulated over time (i.e., the graph can only grow) or by analysing graphs from different points in time.



(a) Both communities recognise each other

(b) Only one community recognises the other

Figure 3: Communities may both recognise each other or one can reveal a “non-social” behaviour

Regarding the phenomena of paradigm shift and merge, there are also other aspects we plan to take into account. One of the communities might show a “non-social” behaviour, simply neglecting the existence and development of new communities. In this case, we expect only one community to cite the other. In contrast, a healthy development would be observed if both communities increasingly cite each other over time. Figure 3 illustrates that difference by using directed edges that indicate the direction of citations. This leads to other important questions, such as what are (un)healthy communities and how to detect that.

Note that the effects are illustrated here in a rather dramatic manner. We expect these phenomena to usually occur alleviated. This means, rather than only one community citing only the other (Figure 3(b)), one community might cite the other in a much more intensive manner. One can see this as this community having more tentacles in other communities. In contrast, certain fields might tend to cite only the “tall” figures visible, even if the community that these figures belong to matures.

3 Roadmap

As outlined before, we focus on citation analysis as the tool of choice for detecting and understanding cross-community effects. It has been shown that this is a reasonable and well-functioning approach for this task. In this section, we provide a general roadmap that defines the different tasks we have to fulfill to achieve our goals. We first discuss how to gather and prepare the citation data needed. Afterwards, we present ways for building social network graphs on top of this data and finally get over to actual methods we aim to apply on the so built networks.

3.1 Data Gathering

First works in the area of citation analysis [White and Griffith, 1980; Gmür, 2003] had to rely on specialised citation databases like the *Social Sciences Citation Index* (SSCI) for gathering the required data. Such databases reveal several disadvantages, such as pruned author lists and only a selection of papers. Luckily, today there are much more citation sources available. Sites like DBLP[4] and Springer[5] are well suited to select a set of seed papers. They also provide ways for selecting high-impact journals and conferences that specifically relate to the chosen communities. Based on this, sites like Google Scholar[6] and CiteULike[7] can be used to extract according citation data, without the need for parsing the chosen papers. The social aspects of, for instance, CiteULike that supports tagging and grouping of works one prefers, further help to identify topics and fields of research. Usually, the raw input data has to be cleaned (different usage of author names and paper titles) and probably pruned to the most significant (most cited) works. We will evaluate if this is also suited for analysing cross-community effects, where we may also be interested in rather small islands of community landscapes.

With most existing approaches, the results of the citation analysis have to be inspected manually in order to deduce meaningful results. [He and Hui, 2002] is one of the first works aiming at automising the whole process. However, the labeling of sub-communities has still to be done on the basis of human inspection. In order to also automate this process, we plan to use a tagging approach. The generated tags can be used to identify and name the found areas of research. To achieve this, one approach is to use already provided keywords and methods from Natural Language Processing. However, we expect this to be too inaccurate and not absolutely satisfying. A second approach that we plan to use is to use input from the communities themselves. For this, a kind of game (similar to the ESP game[8], also adopted by the Google Image Labeler[9]) could be supported. Another idea is to provide an interface for tagging own works ("eating their own dogfood" – a phrase popular in the Semantic Web community). First experiences will show whether these ways for generating tags are sufficient or not.

3.2 Building Citation Graphs

Based on the raw citation data, we will have to build a graph of citation relations. Co-citation analysis has proved to be most suitable for this task. A co-citation between two works (i.e., an edge between two works or authors) is existing if both papers are cited in the same third work. The assumption is that if a co-citation link exists, the works can be regarded as very closely related in the same field of research. Figure 4 illustrates this principle. In this figure, P_3 and P_4 have a stronger co-citation relation than P_3 and P_5 and P_4 and P_5. If P_2 would not cite P_3, P_3 and P_5 would have no co-citation link.

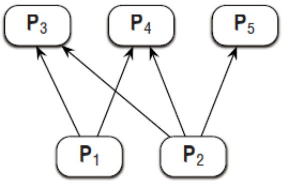


Figure 4: Principle of co-citations [Greene *et al.*, 2008]

There are several different approaches for co-citation analysis. They mainly differ in:

- use a document-based or author-based approach (the nodes in the graph)
- whether to use absolute co-citation counts or relative values like Pearson's correlation coefficient as in [He and Hui, 2002]
- macro vs. micro approach

Document-based analysis provides a more detailed view, but might assign an author to several (sub-)communities. However, we expect the document-based approach as more suitable for our needs, as it focuses on topics rather than geography as the author-based approach does. Usually, relative values can be expected to provide a more realistic view due to their normalising effect. The macro approach focuses on the overall structure of disciplines, whereas the micro approach tries to explain the structure and historical development of single disciplines. See [Gmür, 2003] for a good overview of the different approaches. [Gmür, 2003] also compares different approaches for clustering the citation data. We will evaluate the different methods with respect to their suitability to the special aims and expected phenomena in cross-community analysis.

However, applying only co-citation analysis is not sufficient. To handle aspects mentioned in Section 2, such as only one tall visible figure or direction of citations, we will have to apply pure citation analysis as well. A crucial task is to identify a good mixture of both and how to map the different techniques of co-citation analysis to the case of pure citation analysis. To the best of our knowledge, up to now no work aimed at combining both approaches.

3.3 Analysing Citation Graphs

If we once built the citation graph, methods from social network analysis and graph mining seem to be most promising to analyse it. Maybe one of the most interesting methods is to apply different centrality measures. Eigenvector centrality and degree centrality have been shown to be especially suited [Greene *et al.*, 2008]. In general, we aim to identify bridges, hubs and further nodes of central importance. This might go along with a look on outer-world effects, as there may be bridges that actually belong to a third community. No existing work focuses specifically on effects between predefined communities and the influences of outer-world instances.

To reveal the inter-relationships among authors or works, three different approaches for multivariate analysis have been shown to be especially suited [He and Hui, 2002]:

[4] http://www.informatik.uni-trier.de/ ley/db/index.html

[5] http://www.springer.com

[6] http://scholar.google.com

[7] http://www.citeulike.org

[8] http://www.espgame.org/gwap

[9] http://images.google.com/imagelabeler

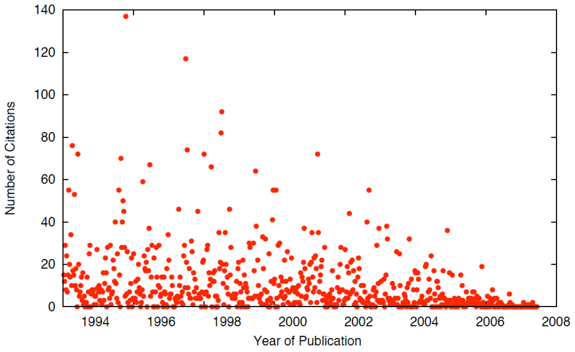


Figure 5: Citation counts over time [Greene *et al.*, 2009]

cluster analysis, multidimensional scaling (MDS) and factor analysis. Cluster analysis builds tree-like cluster representations, either following a top-down or a bottom-up approach. The MDS approach is used to build a map of authors or works, where heavily co-cited authors appear close to each other. MDS is especially suited for visually analysing the communities. More recent approaches for visualising social networks [Henry and Fekete, 2007; Gaudin and Quigley, 2008] will also be applied and evaluated. Factor analysis aims at defining a set of factors that authors contribute to, in their number much smaller than the number of nodes in the graph. Factor analysis has the advantage that it is able to assign authors to more than one factor, i.e., to more than one (sub-)community. [Greene *et al.*, 2008] proposes an interesting combination of hierarchical clustering and factor analysis, called *soft hierarchical clustering*. A detailed evaluation of the different approaches is part of our future work.

It is essential to use citation and co-citation counts to analyse *(i)* the structures in one community as well as *(ii)* the relations between different communities. Otherwise, we will not be able to identify things like a paradigm shift in combination with a paradigm merge as illustrated in Figure 2(b). For instance, looking at Figure 1, one can expect the *Explanation* sub-community as most visible to other communities outside CBR. We are interested in such sub-community effects that come along with cross-community effects.

As mentioned before, we are especially aiming at the analysis of time effects, i.e., the development of citation graphs over time. This involves analysing these graphs accumulated over time as well as in different points of time. It refers to the differences in the link structure as well as to changing positions of authors on an author map [White and Griffith, 1980]. Other time effects must not be ignored as well. For instance, it is natural that older papers are cited more often with time passing. This might be handled by applying the mentioned relative measures. On the other hand, young papers that might play an important role cannot be cited very often, as their visibility just begins to raise. Figure 5 illustrates this by plotting citation counts for papers from the CBR community. To overcome this, [Greene *et al.*, 2008] proposes a back-fitting approach. Later, [Greene *et al.*, 2009] applies clustering based on text-similarity in combination with co-citation analysis. We will evaluate these as well as other approaches for handling that crucial issue. Clearly, more research focusing on such timeline effects is needed. Further aspects we will investigate are possible geographical factors (e.g., the differences between American and European conferences) and the filtering of self-citations. It might also be useful to include factors of availability, such as the time at which online publications are available for certain conferences. A very interesting view is the comparison to methods from computational biology. It can be expected that there exist some parallels, such as communities dying out or surviving, or variants becoming species. One goal is to identify special motifs for certain principles like paradigm shift or (un)healthy communities.

4 Conclusion

In this work, we motivated the interesting possibilities and the impact that the analysis of cross-community effects can have. We illustrated phenomena that can be expected and provided a general view on the process to apply. As an outcome, a major roadmap shows the way and challenges that future works will have to take and face. We believe in a very interesting contribution for scientific research in general. Further, we expect the developed methods and gained experiences to be a valuable contribution for analysing community and cross-community effects in other social networks, which are a dominating factor of today's Web.

References

[Berners-Lee *et al.*, 2001] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, May 2001.

[Gaudin and Quigley, 2008] B. Gaudin and A. J. Quigley. Interactive structural clustering of graphs based on multi-representations. In *Int. Conference Information Visualisation (IV'08)*, pages 227–232, 2008.

[Gmür, 2003] M. Gmür. Co-citation analysis and the search for invisible colleges: A methodological evaluation. *Scientometrics*, 57(1):27–57, 2003.

[Granovetter, 1973] M. Granovetter. The Strength of Weak Ties. *American Journal of Sociology*, 78(6):1360–1380, 1973.

[Greene *et al.*, 2008] D. Greene, J. Freyne, B. Smyth, and P. Cunningham. An Analysis of Research Themes in the CBR Conference Literature. In *European conference on Advances in Case-Based Reasoning (ECCBR'08)*, pages 18–43, 2008.

[Greene *et al.*, 2009] D. Greene, J. Freyne, B. Smyth, and P. Cunningham. An Analysis of Current Trends in CBR Research Using Multi-View Clustering. Technical report, School of Computer Science and Informatics, University College Dublin, Ireland, 2009.

[He and Hui, 2002] Y. He and S. C. Hui. Mining a web citation database for author co-citation analysis. *Inf. Process. Manage.*, 38(4):491–508, 2002.

[Henry and Fekete, 2007] N. Henry and J.-D. Fekete. Nodetrix: a hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1302–1309, 2007.

[Kuhn, 1996] Th. S. Kuhn. *The Structure of Scientific Revolutions*. University Of Chicago Press, December 1996.

[Milgram, 1967] S. Milgram. The small world problem. *Psychology Today*, pages 60–67, 1967.

[White and Griffith, 1980] H. D. White and B. C. Griffith. Author Co-citation: A Literature Measure of Intellectual Structure. *Journal of the American Society for Information Science*, 32:163–171, 1980.

A Framework for Semi-Automatic Development of Rule-based Information Extraction Applications

Peter Kluegl and **Martin Atzmueller** and **Tobias Hermann** and **Frank Puppe**
Department of Computer Science, University of Würzburg, Germany
{pkluegl, atzmueller, hermann, puppe}@informatik.uni-wuerzburg.de

Abstract

For the successful processing and handling of (large scale) document collections, effective information extraction methods are essential. This paper presents a framework for the semi-automatic development of rule-based information extraction applications based on the TEXTMARKER language utilizing machine learning methods. We describe the approach in detail and present the TEXTRULER system as an implementation of the proposed approach.

1 Introduction

Effective methods for information extraction are essential for the (large scale) processing and handling of textual data. In general, information extraction aims to locate specific items in (unstructured) textual documents, e.g., as a first step for more semantic analysis, or for structured data acquisition from text. There exist a variety of automatic methods for information extraction, however, there are also other approaches, e.g., rule-based methods. An implementation of the latter is provided by the TEXTMARKER system wich requires knowledge acquisition. For supporting the developer during rule acquisition step, semi-automatic methods are key techniques.

In this paper, we describe a knowledge-engineering approach incorporating rule-learning methods: For rapid rule capture and prototyping, several rule-learning methods can be applied for acquiring a set of rules that can then be refined later at each level. Machine-learning techniques are applied for acquiring slot and template filler rules for supporting the knowledge engineer when building the set of information extraction rules. The framework supports several machine learning approaches; currently, there are four methods based on the idea of filling single or multi-slot templates specifying the required information. The rules are then either learned, e.g., in a top-down or bottom-up covering manner. The methods are targeted at the TEXTMARKER rule formalization language; TEXTMARKER embeds the proposed framework for rapid rule acquisition using machine-learning techniques.

As an extension of TEXTMARKER, we present the TEXTRULER system as a prototypical implementation of the approach. So far, the approach has been evaluated in several case studies, for example in the medical domain.

The rest of the paper is structured as follows: Section 2 gives a short overview of the TEXTMARKER system and section 3 introduces the semi-automatic development. Then, section 4 concludes with a summary and points at interesting directions for future work.

2 The TEXTMARKER System

The TEXTMARKER system [Atzmueller *et al.*, 2008] is an open source tool[1] for the development of rule-based information extraction applications. The development environment is based on the DLTK[2] framework. It supports the knowledge engineer with a full-featured rule editor, components for the explanation of the rule inference and a build process for generic UIMA Analysis Engines and Type Systems [Ferrucci and Lally, 2004]. Therefore TEXTMARKER components can be easily created and combined with other UIMA components in different information extraction pipelines rather flexibly.

TEXTMARKER applies a specialized rule representation language for the effective knowledge formalization: The rules of the TEXTMARKER language are composed of a list of rule elements that themselves consists of four parts: The mandatory matching condition establishs a connection to the input document by referring to an already existing concept, respectively annotation. The optional quantifier defines the usage of the matching condition similar to regular expressions. Then, additional conditions add constraints to the matched text fragment and additional actions determine the consequences of the rule. Therefore, TEXTMARKER rules match on a pattern of given annotations and, if the additional conditions evaluate true, then they execute their actions, e.g. create a new annotation. If no initial annotations exist, for example, created by another component, a scanner is used to seed simple token annotations contained in a taxonomy.

The TEXTMARKER system provides unique functionality that is usually not found in similar systems. The actions are able to modify the document either by replacing or deleting text fragments or by modifying the view on the document. In this case, the rules ignore some annotations, e.g. HTML markup, or are executed only on the remaining text passages. The knowledge engineer is able to add heuristic knowledge by using scoring rules. Additionally, several language elements common to scripting languages like conditioned statements, loops, procedures, recursion, variables and expressions increase the expressiveness of the language. Rules are able to directly invoke external rule sets or arbitrary UIMA Analysis Engines and foreign libraries can be integrated with the extension mechanism for new language elements.

[1] The source code of the TEXTMARKER project is available at https://sourceforge.net/projects/textmarker/

[2] http://www.eclipse.org/dltk/

3 Semi-Automatic Development

In this section, the semi-automatic development of TEXTMARKER rules is discussed in detail. In the following, we present a comprehensive process model, interesting methods for the automatic acquisition of information extraction rules and the current prototype of the system.

3.1 Process Model

Using the knowledge engineering approach, a knowledge engineer normally writes handcrafted rules to create a domain dependent information extraction application, often supported by a gold standard. When starting the engineering process for the acquisition of the extraction knowledge for possibly new slot or more general for new concepts, machine learning methods are often able to offer support in an iterative engineering process. A conceptual overview of the proposed process model for the semi-automatic development of rule-based information extraction applications is provided in figure 1.

First, a suitable set of documents that contain the text fragments with interesting patterns needs to be selected and annotated with the target concepts. Then, the knowledge engineer chooses and configures the methods for automatic rule acquisition to the best of his knowledge for the learning task: Lambda expressions based on tokens and linguistic features, for example, differ in their application domain from wrappers that process generated HTML pages.

Furthermore, parameters like the window size defining relevant features need to be set to an appropriate level. Before the annotated training documents form the input of the learning task, they are enriched with features generated by the partial rule set of the developed application. The result of the methods, that is the learned rules, are proposed to the knowledge engineer for the extraction of the target concept.

The knowledge engineer has different options to proceed: If the quality, amount or generality of the presented rules is not sufficient, then additional training documents need to be annotated or additional rules have to be handcrafted to provide more features in general or more appropriate features. Rules or rule sets of high quality can be modified, combined or generalized and transfered to the rule set of the application in order to support the extraction task of the target concept. In the case that the methods did not learn reasonable rules at all, the knowledge engineer proceeds with writing handcrafted rules.

Having gathered enough extraction knowledge for the current concept, the semi-automatic process is iterated and the focus is moved to the next concept until the development of the application is completed.

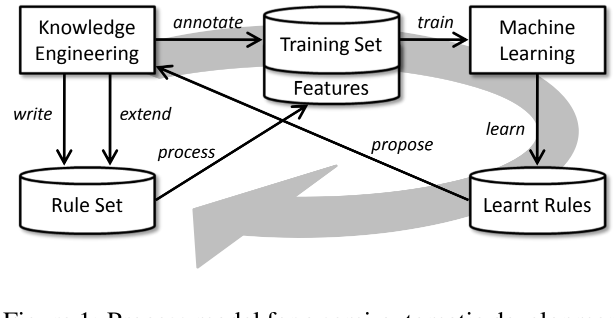


Figure 1: Process model for a semi-automatic development of rule-based information extraction applications.

3.2 Methods

In order to choose appropriate algorithms from the variety of different machine learning techniques for information extraction applications, the following important criteria need to be considered:

- The document type the system operates on.
- Supervised or unsupervised learning strategy.
- Black-box or white-box models.
- Available description of the algorithm.

Since the methods are used in cooperation with the knowledge engineer, rule-based supervised white-box methods fit the described process model best. The goal is to assemble a heterogenous set of methods with techniques for different document types, different learning strategies (e.g. top-down vs. bottom-up) and different rule-representations. The following learning methods all utilize annotated training documents, and have been chosen for further investigation:

BWI

BWI (Boosted Wrapper Induction) [Freitag and Kushmerick, 2000] uses boosting techniques to improve the performance of simple pattern matching single-slot boundary wrappers (*boundary detectors*). Two sets of detectors are learned: the "fore" and the "aft" detectors. Weighted by their confidences and combined with a slot length histogram derived from the training data they can classify a given pair of boundaries within a document. BWI can be used for structured, semi-structured and free text. The patterns are token-based with special wildcards for more general rules.

CRYSTAL

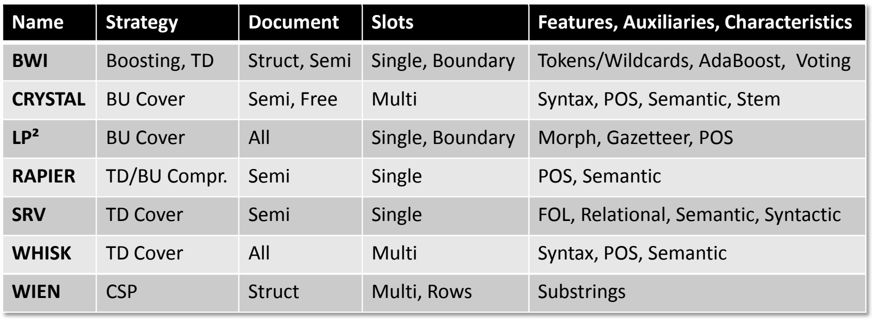
CRYSTAL [Soderland, 1996] learns free-text multi-slot extraction rules named *concept definitions*, which are build of lexical, semantic and syntactic constraints. They operate on sentences or syntactic constituents that are created by a sentence analyzer. Concept definitions are induced in a bottom-up covering manner by generalizing most specific seed rules created from uncovered training instances. A seed rule is merged with the *most similar concept definition* of the initial rule base.

LP^2

This method [Ciravegna, 2003] operates on all three kinds of documents. It learns separate rules for the beginning and the end of a single slot. So called *tagging rules* insert boundary SGML tags and additionally induced *correction rules* shift misplaced tags to their correct positions in order to improve precision. The learning strategy is a bottom-up covering algorithm. It starts by creating a specific seed instance with a window of w tokens to the left and right of the target boundary and searches for the best generalization. Other linguistic NLP-features can be used in order to generalize over the flat word sequence.

RAPIER

RAPIER [Califf and Mooney, 2003] induces single slot extraction rules for semi-structured documents. The rules consist of three patterns: a pre-filler, a filler and a post-filler pattern. Each can hold several constraints on tokens and their according POS-tag- and semantic information. The algorithm uses a bottom-up compression strategy, starting with a most specific seed rule for each training instance. This initial rule base is compressed by randomly selecting



Name	Strategy	Document	Slots	Features, Auxiliaries, Characteristics
BWI	Boosting, TD	Struct, Semi	Single, Boundary	Tokens/Wildcards, AdaBoost, Voting
CRYSTAL	BU Cover	Semi, Free	Multi	Syntax, POS, Semantic, Stem
LP²	BU Cover	All	Single, Boundary	Morph, Gazetteer, POS
RAPIER	TD/BU Compr.	Semi	Single	POS, Semantic
SRV	TD Cover	Semi	Single	FOL, Relational, Semantic, Syntactic
WHISK	TD Cover	All	Multi	Syntax, POS, Semantic
WIEN	CSP	Struct	Multi, Rows	Substrings

Figure 2: Overview of the adressed methods. (TD = Top-Down, BU = Bottom-Up)

rule pairs and search for the best generalization. Considering two rules, the least general generalization (LGG) of the slot fillers are created and specialized by adding rule items to the pre- and post-filler until the new rules operate well on the training set. The best of the k rules (k-beam search) is added to the rule base and all empirically subsumed rules are removed.

SRV

SRV [Freitag, 2000] uses single-slot *first order logic* (FOL) rules to classify a given text fragment. It is an ILP system based on FOIL and is suitable for all three kinds of documents dependent on the used feature set. Rules are created in a top-down covering manner from positive and negative instances by starting with a general rule and adding literals until the rule operates well on the training set. It uses a feature-set containing simple attribute-value features and relational features.

WHISK

Another multi-slot method is WHISK [Soderland *et al.*, 1999]. It can operate on all three kinds of documents and learns single- or multi-slot rules looking similar to regular expressions. The top-down covering algorithm begins with the most general rule and specializes it by adding single rule terms until the rule makes no errors on the training set. Domain specific classes or linguistic information obtained by a syntactic analyzer can be used as additional features. The exact definition of a rule term (e.g. a token) and of a problem instance (e.g. a whole document or a single sentence) depends on the operating domain and document type.

WIEN

WIEN [Kushmerick *et al.*, 1997] is the only method listed here that operates on highly structured texts only. It induces so called *wrappers* that anchor the slots by their structured context around them. The HLRT (*head left right tail*) wrapper class for example can determine and extract several multi-slot-templates by first separating the important information block from unimportant head and tail portions and then extracting multiple data rows from table like data structures from the remaining document. Inducing a wrapper is done by solving a CSP for all possible pattern combinations from the training data.

Figure 2 gives a short overview of the adressed methods by summarizing the strategy of the algorithm, the allowed document types, the extraction output and the commonly used features, auxiliary methods or further characteristics.

3.3 The TEXTRULER System

The prototype of the TEXTRULER system was developed in [Hermann, 2009] and is currently being extended. Figure 3 shows a screenshot of the TEXTMARKER system and the integrated TEXTRULER system. Different components for the creation of labeled training documents, configuration of the selected methods and visualization of the learned rules allow the usage of the presented semi-automatic process model. Currently, prototypes of four of the six presented methods are implemented for the TEXTMARKER language and three[3] of them are rated extracting headlines for diagnoses, therapies and examinations in medical discharge letters. The following criteria adress the usefulness of the methods for a knowlegde engineer:

Comprehensibility

The comprehensibility of the learned rules is essential for the introspection, selection and further engineering of the new rules. The structure and length should not conceal the coherences between the patterns in the input document and the used features and language constructs of the rules.

Extensibility

The knowledge engineer should be able to extend and optimize the proposed rules by adapting and generalizing the language elements and their used features. A transfer of the rules to other domains and the possible improvement especially by the human way of thinking are rated.

Integratability

The learned rules need to be integrated in the existing rule set. This criteria weights the straightforwardness of the integration and transferable constructs, e.g., rules.

Usage of Features

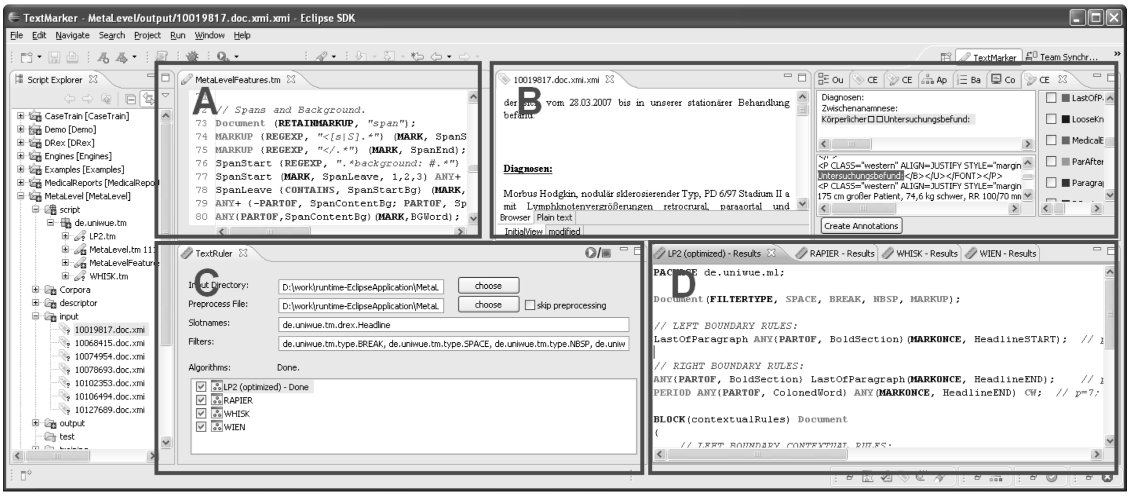
The methods' usage of the given features and in particular the additional features of the extended rule set is of central interest for an iterative knowledge engineering. Therefore, not only the included features, but also their types and concepts for further improvements are rated.

Result, Performance

The time spent on learning and the amount of learned rules should be adequate. Finally, the extraction speed and accuracy of the rules in the focused domain are rated.

The results of the qualitative rating of the methods' current implementations are listed in figure 4. The boundary rule representation of LP^2 impairs the readability and the further engineering. The lack of integrated features and the

[3] Only LP^2, WHISK, and RAPIER have been rated, since WIEN is not applicable for the learning task.



- CaseTrain [CaseTrain]
- Demo [Demo]
- DRex [DRex]
- Engines [Engines]
- Examples [Examples]
- MedicalReports [MedicalRepo
- MetaLevel [MetaLevel]
  - script
    - de.uniwue.tm
      - LP2.tm
      - MetaLevel.tm 11
      - MetaLevelFeature
      - WHISK.tm
  - Corpora
  - descriptor
  - input
    - 10019817.doc.xmi
    - 10068415.doc.xmi
    - 10074954.doc.xmi
    - 10078693.doc.xmi
    - 10102353.doc.xmi
    - 10106494.doc.xmi
    - 10127689.doc.xmi
  - output
  - test

```
// Spans and Background.
Document (RETAINMARKUP, "span");
MARKUP {REGEXP, "<[s|S].*"} (MARK, SpanS
MARKUP {REGEXP, "</.*"} (MARK, SpanEnd);
SpanStart {REGEXP, ".*background: #.*"}
SpanStart (MARK, SpanLeave, 1,2,3) ANY+
SpanLeave {CONTAINS, SpanStartBg} (MARK,
ANY+ {-PARTOF, SpanContentBg; PARTOF, Sp
ANY{PARTOF,SpanContentBg}(MARK,BGWord);
```

der vom 28.03.2007 bis in unserer stationärer Behandlung befand.

Diagnosen:

Morbus Hodgkin, nodulär sklerosierender Typ, PD 6/97 Stadium II a mit Lymphknotenvergrößerungen retrocrural, paraaortal und

Diagnosen:
Zwischenanamnese:
Körperlicher□□Untersuchungsbefund:

```
<P CLASS="western" ALIGN=JUSTIFY STYLE="margin
Untersuchungsbefund:</B></U></FONT></P>
<P CLASS="western" ALIGN=JUSTIFY STYLE="margin
175 cm großer Patient, 74,6 kg schwer, RR 100/70 mn
```

- □ LastOfP.
- □ LooseKn
- □ MedicalE
- □ ParAfter
- □ Paragra

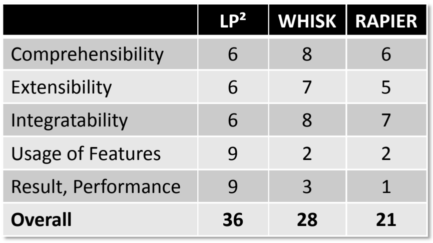
Input Directory: D:\work\runtime-EclipseApplication\MetaL [choose]
Preprocess File: D:\work\runtime-EclipseApplication\MetaL [choose] □ skip preprocessing
Slotnames: de.uniwue.tm.drex.Headline
Filters: de.uniwue.tm.type.BREAK, de.uniwue.tm.type.SPACE, de.uniwue.tm.type.NBSP, de.uniw
Algorithms: Done.

- ☑ LP2 (optimized) - Done
- ☑ RAPIER
- ☑ WHISK
- ☑ WIEN

```
PACKAGE de.uniwue.ml;

Document(FILTERTYPE, SPACE, BREAK, NBSP, MARKUP);

// LEFT BOUNDARY RULES:
LastOfParagraph ANY{PARTOF, BoldSection}(MARKONCE, HeadlineSTART);  // p

// RIGHT BOUNDARY RULES:
ANY{PARTOF, BoldSection} LastOfParagraph(MARKONCE, HeadlineEND);    // p
PERIOD ANY{PARTOF, ColonedWord} ANY(MARKONCE, HeadlineEND) CW;  // p=7;

BLOCK(contextualRules) Document
{
    // LEFT BOUNDARY CONTEXTUAL RULES:
```

Figure 3: The TEXTRULER System: (A) Part of the TEXTMARKER development environment. (B) Editor for creating labeled training documents. (C) Control panel of the TEXTRULER system for the selection of methods and their parameters. (D) Results of the current semi-automatic development iteration.

necessary time spent on learning of RAPIER and WHISK prevent their practical usage. However, both methods are able to gain on LP^2, if their performance and features are improved. For a detailed feedback see [Hermann, 2009].



	LP²	WHISK	RAPIER
Comprehensibility	6	8	6
Extensibility	6	7	5
Integratability	6	8	7
Usage of Features	9	2	2
Result, Performance	9	3	1
Overall	**36**	**28**	**21**

Figure 4: Qualitative rating of the current implementations in the TEXTRULER framework. (1 = weak, 10 = good)

4 Conclusions

In this paper, we have described a framework for the semi-automatic development of rule-based information extraction applications. We have presented the TEXTRULER and the TEXTMARKER systems implementing the presented approach. The TEXTRULER system provides the machine learning methods and is embedded into the TEXTMARKER system. The current prototypical implementations have been rated using a case study in the medical domain.

For future work, the implemented methods need to be improved, especially the used TextMarker language constructs and the features that are applied for annotation and information extraction. Additionally, we want to implement a more comprehensive set of learning methods covering all the discussed techniques. Furthermore, we also aim to develop novel methods that are optimized for TEXTMARKER, especially using the provided language constructs.

Acknowledgements

This work has been partially supported by the German Research Council (DFG) under grant Pu 129/8-2.

References

[Atzmueller *et al.*, 2008] Martin Atzmueller, Peter Kluegl, and Frank Puppe. Rule-Based Information Extraction for Structured Data Acquisition using TextMarker. In *Proc. of the LWA-2008 (KDML Track)*, pages 1–7, 2008.

[Califf and Mooney, 2003] Mary Elaine Califf and Raymond J. Mooney. Bottom-up Relational Learning of Pattern Matching Rules for Information Extraction. *Journal of Machine Learning Research*, 4:177–210, 2003.

[Ciravegna, 2003] F. Ciravegna. (LP)^2, Rule Induction for Information Extraction Using Linguistic Constraints. Technical Report CS-03-07, Department of Computer Science, University of Sheffield, Sheffield, 2003.

[Ferrucci and Lally, 2004] David Ferrucci and Adam Lally. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Nat. Lang. Eng.*, 10(3-4):327–348, 2004.

[Freitag and Kushmerick, 2000] Dayne Freitag and Nicholas Kushmerick. Boosted Wrapper Induction. In *AAAI/IAAI*, pages 577–583, 2000.

[Freitag, 2000] Dayne Freitag. Machine Learning for Information Extraction in Informal Domains. *Machine Learning*, 39(2):169–202, 2000.

[Hermann, 2009] Tobias Hermann. Semi-Automatic Development of Information Extraction Systems using Machine Learning Techniques (in german). Master's thesis, Univ. Wuerzburg, Dep. for Computer Science VI, 2009.

[Kushmerick *et al.*, 1997] N. Kushmerick, D. Weld, and B. Doorenbos. Wrapper Induction for Information Extraction. In *Proc. IJC Artificial Intelligence*, 1997.

[Soderland *et al.*, 1999] Stephen Soderland, Claire Cardie, and Raymond Mooney. Learning Information Extraction Rules for Semi-Structured and Free Text. In *Machine Learning*, volume 34, pages 233–272, 1999.

[Soderland, 1996] S. G. Soderland. Learning Text Analysis Rules for Domain-specific Natural Language Processing. Technical report, University of Massachusetts, Amherst, MA, USA, 1996.

Meta-Level Information Extraction

Peter Kluegl and **Martin Atzmueller** and **Frank Puppe**
University of Würzburg,
Department of Computer Science VI
Am Hubland, 97074 Würzburg, Germany
{pkluegl, atzmueller, puppe}@informatik.uni-wuerzburg.de

Abstract

This paper presents a novel approach for meta-level information extraction (IE). The common IE process model is extended by utilizing transfer knowledge and meta-features that are created according to already extracted information. We present two real-world case studies demonstrating the applicability and benefit of the approach and directly show how the proposed method improves the accuracy of the applied information extraction technique.

1 Introduction

While structured data is readily available for information and knowledge extraction, unstructured information, for example, obtained from a collection of text documents cannot be directly utilized for such purposes. Since there is significantly more unstructured (textual) information than structured information e.g., obtained by structured data acquisition information extraction (IE) methods are rather important. This can also be observed by monitoring the latest developments concerning IE architectures, for example *UIMA* [Ferrucci and Lally, 2004] and the respective methods, e.g., conditional random fields (CRF), support vector machines (SVM), and other (adaptive) IE methods, cf., [McCallum and Li, 2003; Turmo *et al.*, 2006; Li *et al.*, 2008].

Before IE is applied, first an IE model is learned and generated in the learning phase. Then, the IE process model follows a standard approach depicted in Figure 1: As the first step of the process itself, the applicable features are extracted from the document. In some cases, a (limited form of) knowledge engineering is used for tuning the relevant features of the domain. Finally, the generated model is applied on the data such that the respective IE method selects or classifies the relevant text fragments and extracts the necessary information.

With respect to the learning phase, usually machine-learning related approaches like candidate classification, windowing, and markov models are used. Often SVMs or CRFs are applied for obtaining the models. The advantages of CRFs are their relation to the sequence labeling problem, while they do not suffer from the dependencies between the features. However, a good feature selection step is still rather important. The advantages of SVMs are given by their automatic ranking of the input features and their ability to handle a large number of features. Therefore, less knowledge engineering is necessary. However, the IE task can also be implemented by knowledge engineering approaches applying rules or lambda expressions. In the case studies we present a rule-based approach that is quite effective compared to the standard approaches.

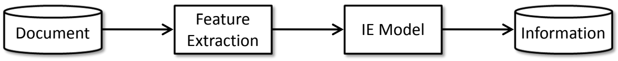


Figure 1: Common Process Model for Information Extraction.

Specifically, this paper proposes extensions to the common IE approach, such that meta-level features that are generated during the IE process can be utilized: The creation of the meta-level features is based on the availability of already extracted information that is applied in a feedback loop. Then repetitive information like structural repetitions can be processed further by utilizing transfer knowledge. Assuming that a document, for example, is written by a single author, then it is probably the case that the same writing and layout style is used for all equivalent structures. We present two real-world case studies demonstrating the applicability and benefit of the approach and show how the proposed method improves the accuracy of the applied information extraction technique.

The rest of the paper is structured as follows: Section 2 presents the proposed novel process model for information extraction extending the standard process. We first motivate the concrete problem setting before we discuss the extensions in detail and specifically the techniques for meta-level information extraction. After that, Section 3 presents two real-world case studies: We demonstrate the applicability of the presented approach, for which the results directly indicate its benefit. Next, we discuss related work. Finally, Section 4 concludes with a summary of the presented approach and points at interesting directions for future research.

2 Meta-Level Information Extraction

In the following, we first motivate the proposed approach by presenting two examples for which the commonly applied standard process model is rather problematic. Next, we present the process model for meta-level information extraction and discuss its elements and extensions in detail.

2.1 Problem Statement

To point out certain flaws of the standard process, we discuss examples concerning information extraction from curricula vitae (CV) and from medical discharge letters. Both application domains indicate certain problems of the common process model for information extraction and lead to following claim:

> **Claim:** Using already extracted information for further information extraction can often account for missing or ambiguous features and increase the accuracy in domains with repetitive structure(s).

Example 1: CVs

For the extraction from CV documents, a predefined template with slots for start time, end time, title, company and description is filled with the corresponding text fragments. The text segments describing experiences or projects are used to identify a template. Then, the slots of the templates are extracted. Often the company can be identified using simple features, e.g., common suffixes, lists of known organizations or locations. Yet, these word lists cannot be exhaustive, and are often limited for efficiency reasons, e.g., for different countries. This can reduce the accuracy of the IE model, e.g., if the employee had been working in another country for some time.

Humans solve these problems of missing features, respectively of unknown company names, by transferring already 'extracted' patterns and relations. If the company, for example, was found in the third line of ninety percent of all project sections, then it is highly probable that an 'unclear' section contains a company name in the same position.

Example 2: Medical discharge letters

Medical discharge letters contain, for example, the observations, the history, and the diagnoses of the patient. For IE, different sections of the letter need to be identified: The headlines of a section cannot only help to create a segmentation of the letter, but also provide hints what kind of sections and observations are present. Since there are no restrictions, there is a variety of layout structure; Figure 2 shows some examples: Whereas the headlines in (A) are represented using a table, (C+D) use bold and underlined. However, (B) and (F) color the headlines' background and use bold and underlined for subheaders.

Some physicians writing the discharge letters apply layout features only to emphasize results and not for indicating a headline. It is obvious, that a classification model can then face significant problems, if the relation between features and information differs for each input document. In contrast, humans are able to identify common headlines 'semantically' using the contained content (words). Then, they transfer such significant features to other text fragments and extract headlines with a similar layout.

2.2 Process Model

The human behavior solving the flaws of the common IE process model seems straight forward, yet its formalization using rules or statistical models is quite complex. We approach this challenge by proposing an extended process model, shown in figure 3: Similar to the common IE process model, features are extracted from the input document and are used by a static IE model to identify the information. Expectations or self-criticism can help to identify highly confident information and relevant meta-features.

Transfer knowledge is responsible for the projection or comparison of the given meta-features. The meta-features and transfer knowledge elements make up the dynamic IE model and are extended in an incremental process. The elements of the process model are discussed in more detail in the following:

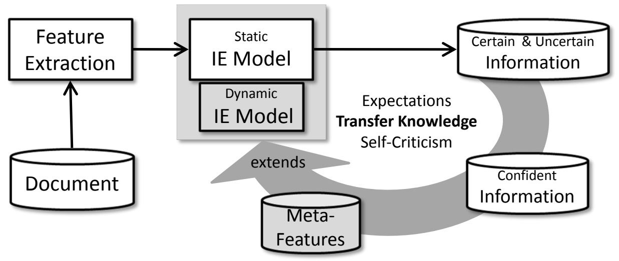


Figure 3: Extended process model with meta-features and transfer knowledge.

Meta-Features

Relations between features and information, respectively patterns, are explicitly implemented by meta-features. These are not only created for the extracted information, but also for possible candidates. A simple meta-feature, for example for the extraction of headlines, states that the bold feature indicates a headline in this document.

Expectations and Self-Criticism

Since even only a single incorrect information can lead to a potentially high number of incorrect information, the correctness and confidence of an information is essential for the meta-level information extraction. There are two ways to identify an information suitable for the extraction of meta-features. If the knowledge engineer already has some assumptions about the content of the input documents, especially on the occurrence of certain information, then these expectations can be formalized in order to increase the confidence of the information. In the absence of expectations, self-criticism of the IE model using features or a confidence value can highlight a suitable information. Furthermore, self-criticism can be used to reduce the incorrect transfer of meta-features by rating newly identified information.

Transfer Knowledge

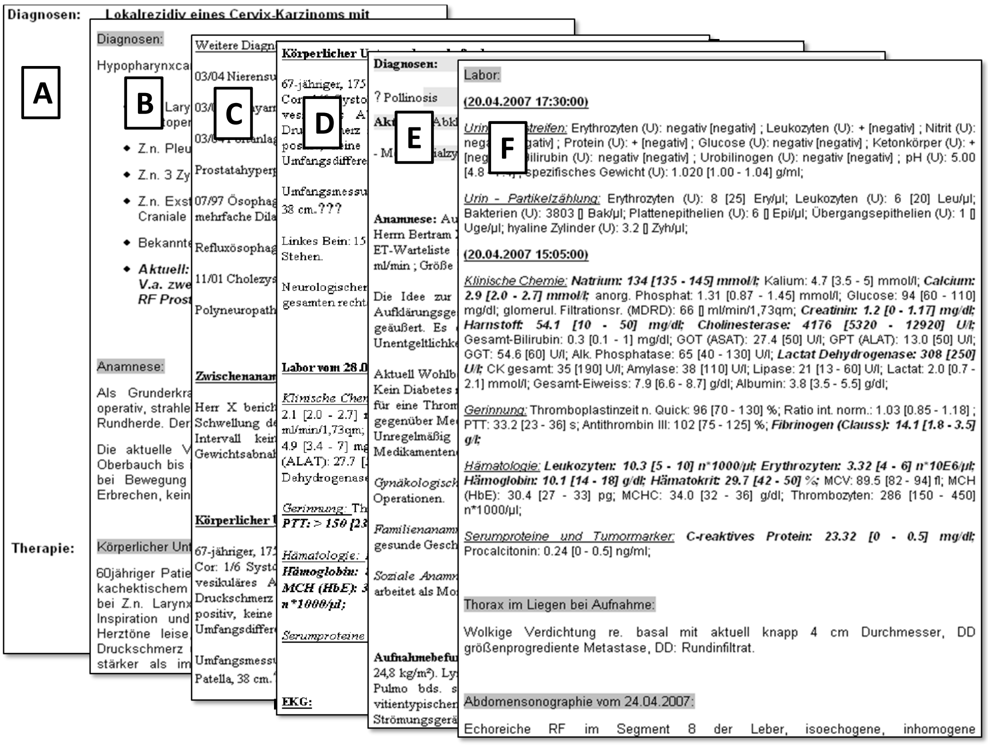
The transfer knowledge models the human behavior in practice and can be classified in three categories: *Agglomeration* knowledge processes multiple meta-features and creates new composed meta-features. Then, *projection* knowledge defines the transfer of the meta-features to possible candidates of new information. *Comparison* knowledge finally formalizes how the similarity of the meta-features of the original information and a candidate information is calculated. The usage of these different knowledge types in an actual process depends on the kind of repetitive structures and meta-features. In section 3, specific examples of these elements are explained in the context of their application.

3 Case Studies

For a demonstration of the applicability and benefit of the approach, the two subtasks of the IE applications introduced earlier are addressed. The meta-level approach is realized with the rule-based TextMarker system and the statistical natural language processing toolkit ClearTK [Ogren *et al.*, 2008] is used for the supervised machine learning methods CRF[1] and SVM[2]. The three methods operate in the same architecture (UIMA) and process the identical

[1] Mallet (`http://mallet.cs.umass.edu/`)

[2] SVMLight (`http://svmlight.joachims.org/`)


A
B
C
D
E
F
Diagnosen:
Therapie:
Anamnese:
Körperlicher Un
Weitere Diagn
Zwischenanam
Labor vom 28.0
Diagnosen:

Labor:

(20.04.2007 17:30:00)

Urin streifen: Erythrozyten (U): negativ [negativ] ; Leukozyten (U): + [negativ] ; Nitrit (U): nega gativ] ; Protein (U): + [negativ] ; Glucose (U): negativ [negativ] ; Ketonkörper (U): + [neg Bilirubin (U): negativ [negativ] ; Urobilinogen (U): negativ [negativ] ; pH (U): 5.00 [4.8 spezifisches Gewicht (U): 1.020 [1.00 - 1.04] g/ml;

Urin - Partikelzählung: Erythrozyten (U): 8 [25] Ery/µl; Leukozyten (U): 6 [20] Leu/µl; Bakterien (U): 3803 [] Bak/µl; Plattenepithelien (U): 6 [] Epi/µl; Übergangsepithelien (U): 1 [] Uge/µl; hyaline Zylinder (U): 3.2 [] Zyh/µl;

(20.04.2007 15:05:00)

Klinische Chemie: ***Natrium: 134 [135 - 145] mmol/l;*** Kalium: 4.7 [3.5 - 5] mmol/l; ***Calcium: 2.9 [2.0 - 2.7] mmol/l;*** anorg. Phosphat: 1.31 [0.87 - 1.45] mmol/l; Glucose: 94 [60 - 110] mg/dl; glomerul. Filtrationsr. (MDRD): 66 [] ml/min/1,73qm; ***Creatinin: 1.2 [0 - 1.17] mg/dl; Harnstoff: 54.1 [10 - 50] mg/dl; Cholinesterase: 4176 [5320 - 12920] U/l;*** Gesamt-Bilirubin: 0.3 [0.1 - 1] mg/dl; GOT (ASAT): 27.4 [50] U/l; GPT (ALAT): 13.0 [50] U/l; GGT: 54.6 [60] U/l; Alk. Phosphatase: 65 [40 - 130] U/l; ***Lactat Dehydrogenase: 308 [250] U/l;*** CK gesamt: 35 [190] U/l; Amylase: 38 [110] U/l; Lipase: 21 [13 - 60] U/l; Lactat: 2.0 [0.7 - 2.1] mmol/l; Gesamt-Eiweiss: 7.9 [6.6 - 8.7] g/dl; Albumin: 3.8 [3.5 - 5.5] g/dl;

Gerinnung: Thromboplastinzeit n. Quick: 96 [70 - 130] %; Ratio int. norm.: 1.03 [0.85 - 1.18] ; PTT: 33.2 [23 - 36] s; Antithrombin III: 102 [75 - 125] %; ***Fibrinogen (Clauss): 14.1 [1.8 - 3.5] g/l;***

Hämatologie: ***Leukozyten: 10.3 [5 - 10] n*1000/µl; Erythrozyten: 3.32 [4 - 6] n*10E6/µl; Hämoglobin: 10.1 [14 - 18] g/dl; Hämatokrit: 29.7 [42 - 50] %;*** MCV: 89.5 [82 - 94] fl; MCH (HbE): 30.4 [27 - 33] pg; MCHC: 34.0 [32 - 36] g/dl; Thrombozyten: 286 [150 - 450] n*1000/µl;

Serumproteine und Tumormarker: ***C-reaktives Protein: 23.32 [0 - 0.5] mg/dl;*** Procalcitonin: 0.24 [0 - 0.5] ng/ml;

Thorax im Liegen bei Aufnahme:

Wolkige Verdichtung re. basal mit aktuell knapp 4 cm Durchmesser, DD größenprogrediente Metastase, DD: Rundinfiltrat.

Abdomensonographie vom 24.04.2007:

Echoreiche RF im Segment 8 der Leber, isoechogene, inhomogene

Figure 2: Examples of different headlines in medical discharge letters.

features. The same documents are applied for the training and test phase of the machine learning approaches and intentionally no k-fold cross evaluation is used, since it is hardly applicable for the knowledge engineering approach. Yet, the selected features do not amplify overfitting, e.g., no stem information is used. The evaluation of the SVM did not return reasonable values, probably because of the limited amount of documents and features in combination with the selected kernel method. Therefore, only results of the meta-level approach and CRF are presented using the F1-measure.

3.1 The TextMarker System

The TEXTMARKER system[3] is a rule-based tool for information extraction and text processing tasks [Atzmueller *et al.*, 2008]. It provides a full-featured development environment based on the DLTK framework[4] and a build process for UIMA Type Systems and generic UIMA Analysis Engines [Ferrucci and Lally, 2004]. Different components for rule explanation and test-driven development [Kluegl *et al.*, 2008] facilitate the knowledge engineering of rule-based information extraction components. The basic idea of the TEXTMARKER language is similar to JAPE [Cunningham *et al.*, 2000]: rules match on combinations of predefined types of annotations and create new annotations. Furthermore, the TEXTMARKER language provides an extension mechanism for domain dependent language elements, several scripting functionalities and a dynamic view on the document. Due to the limited space, we refer to [Atzmueller *et al.*, 2008; Kluegl *et al.*, 2008] for a detailed description of the system.

3.2 CVs

In this case study, we evaluate a subtask of the extraction of CV information: Companies in a past work experience of a person, respectively the employer. The corpus contains only 15 documents with 72 companies. The selected features consists of already extracted slots, layout information, simple token classes and a list of locations of one country. The meta-features are based on the position of confident information dependent on the layout and in relation to other slots. Agglomeration knowledge uses these meta-features to formalize a pattern of the common appearance of the companies. Then, projection knowledge uses this pattern to identify new information, that is rated by rules for self-criticism. In Figure 4, the results of the evaluation are listed. The meta-level approach achieved a F1-measure of 97.87% and the CRF method reached 75.00%. The low recall value of the CRF is caused by the limited amount of available features. However, the meta-level approach was able to compensate for this loss using the meta-features.

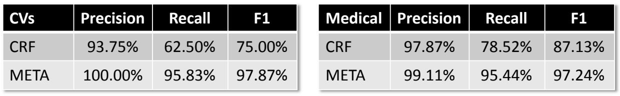
3.3 Medical discharge letters

A subtask of the extraction of information from medical discharge letters is the recognition of headlines. In order to evaluate the approaches we use a corpus with 141 documents and 1515 headlines. The extracted features consist mainly of simple token classes and layout information, e.g., bold, underlined, italic and freeline. In this case study, the expectation to find a *Diagnose* or *Anamnese* headline is used to identify a confident information. Then, meta-features describing its actual layout are created and transferred by projection knowledge. Finally, comparison knowledge is used to calculate the similarity of the layout of the confident information and a candidate for a headline. The results of the evaluation are shown in figure 4: The meta-level approach was evaluated with 97.24% and

[3] http://textmarker.sourceforge.net/

[4] http://www.eclipse.org/dltk/

the CRF method achieved a F1-measure of 87.13%. CRF extracted the same headlines as the meta-level approach in many documents. However, the conflicting layout styles of the some authors caused, as expected, a high number of false negative errors resulting in a lower recall value.



CVs	Precision	Recall	F1
CRF	93.75%	62.50%	75.00%
META	100.00%	95.83%	97.87%

Medical	Precision	Recall	F1
CRF	97.87%	78.52%	87.13%
META	99.11%	95.44%	97.24%

Figure 4: Results of the CVs and medical discharge letters evaluation

3.4 Related Work and Discussion

In the case studies, we have seen that the proposed approach performs very promising and achieves considerably better accuracy measures than the approach using machine learning techniques. The machine learning methods would potentially perform better using more or 'better' features, however, the same is true for the meta-level IE approach. The approach is not only very effective but also rather efficient, since the proposed approach required only about 1-2 hours for formalizing the necessary meta-features and transfer knowledge, significantly less time than the time spent for the annotation of the examples.

To the best of the authors' knowledge, the approach is novel in the IE community and application. However, similar ideas to the core idea of transferring features have been addressed in the feature construction and inductive logic programming community, e.g., [Flach and Lavrac, 2000]. However, in this context there is no direct 'feedback' according to a certain process, and also no distinction between meta-features and transfer knowledge that is provided by the presented approach. According to the analogy of human reasoning, it is often easier to formalize each knowledge element separately. Especially in information extraction, there are approaches using extracted information in a meta-learning process, e.g., [Sigletos *et al.*, 2003]. However, compared to our approach no meta-features dependent on extracted information and no transfer knowledge is used. The proposed approach is able to adapt to peculiarities of certain authors of the documents, similarly to the adaptation phase of common speech processing and speech understanding systems.

4 Conclusions

In this paper, we have presented a meta-level information extraction approach that extends the common IE process model by including meta-level features. These meta-features are created using already extracted information, e.g., given by repetitive structural constructs of the present feature space. We have described a general model for the application of the presented approach, and we have demonstrated its benefit in two case studies utilizing a rule-based system for information extraction.

For future work, the exchange of transfer knowledge and meta-features between documents can further enrich the process model in specific domains. We plan to extend the approach in order to incorporate the automatic acquisition of transfer knowledge. Techniques from inductive logic programming [Thomas, 2005] can potentially provide helpful methods and support the knowledge engineer to automatically acquire the needed transfer and meta knowledge. For the automatic acquisition, self criticism capabilities and the inclusion of the expectations of the developer are essential. Finally, we are also planning to combine the approach with learning methods, like SVM and CRF.

Acknowledgements

This work has been partially supported by the German Research Council (DFG) under grant Pu 129/8-2.

References

[Atzmueller *et al.*, 2008] Martin Atzmueller, Peter Kluegl, and Frank Puppe. Rule-Based Information Extraction for Structured Data Acquisition using TextMarker. In *Proc. of the LWA-2008 (KDML Track)*, pages 1–7, 2008.

[Cunningham *et al.*, 2000] H. Cunningham, D. Maynard, and V. Tablan. JAPE: A Java Annotation Patterns Engine (Second Edition). Research Memorandum CS–00–10, University of Sheffield, 2000.

[Ferrucci and Lally, 2004] David Ferrucci and Adam Lally. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Nat. Lang. Eng.*, 10(3-4):327–348, 2004.

[Flach and Lavrac, 2000] Peter A. Flach and Nada Lavrac. The Role of Feature Construction in Inductive Rule Learning. In Luc De Raedt and Stefan Kramer, editors, *Proc. ICML2000 Workshop on Attribute-Value and Relational Learning: crossing the boundaries*, pages 1–11. 17th Int. Conf. on Machine Learning, July 2000.

[Kluegl *et al.*, 2008] Peter Kluegl, Martin Atzmueller, and Frank Puppe. Test-Driven Development of Complex Information Extraction Systems using TextMarker. In *KESE at KI 2008*, 2008.

[Li *et al.*, 2008] Dingcheng Li, Guergana Savova, and Karin Kipper-Schuler. Conditional Random Fields and Support Vector Machines for Disorder Named Entity Recognition in Clinical Texts. In *Proc. of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 94–95, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[McCallum and Li, 2003] Andrew McCallum and Wei Li. Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons. In *Proc. of the seventh conf. on nat. lang. learning at HLT-NAACL 2003*, pages 188–191, Morristown, NJ, USA, 2003.

[Ogren *et al.*, 2008] P. V. Ogren, P. G. Wetzler, and S. Bethard. ClearTK: A UIMA Toolkit for Statistical Natural Language Processing. In *UIMA for NLP workshop at LREC*, 2008.

[Sigletos *et al.*, 2003] Georgios Sigletos, Georgios Paliouras, Constantine D. Spyropoulos, and Takis Stamatopoulos. Meta-Learning beyond Classification: A Framework for Information Extraction from the Web. In *Proc. of the Workshop on Adaptive Text Extraction and Mining. The 14th Euro. Conf. on Machine Learning and the 7th Euro. Conf. on PKDD*, 2003.

[Thomas, 2005] Bernd Thomas. *Machine Learning of Information Extraction Procedures - An ILP Approach*. PhD thesis, Universität Koblenz-Landau, 2005.

[Turmo *et al.*, 2006] Jordi Turmo, Alicia Ageno, and Neus Català. Adaptive Information Extraction. *ACM Comput. Surv.*, 38(2):4, 2006.

Extracting Human Goals from Weblogs

Mark Kröll
Graz University of Technology
Inffeldgasse 21a
8010 Graz, Austria
mkroell@tugraz.at

Markus Strohmaier
Graz University of Technology and Know-Center
Inffeldgasse 21a
8010 Graz, Austria
markus.strohmaier@tugraz.at

Abstract

Knowledge about human goals has been found to be an important kind of knowledge for a range of challenging problems, such as goal recognition from peoples' actions or reasoning about human goals. Necessary steps towards conducting such complex tasks involve (i) acquiring a broad range of human goals and (ii) making them accessible by structuring and storing them in a knowledge base. In this work, we focus on extracting goal knowledge from weblogs, a largely untapped resource that can be expected to contain a broad variety of human goals. We annotate a small sample of weblogs and devise a set of simple lexico-syntactic patterns that indicate the presence of human goals. We then evaluate the quality of our patterns by conducting a human subject study. Resulting precision values favor patterns that are not merely based on part-of-speech tags. In future steps, we intend to improve these preliminary patterns based on our observations.

1 Knowledge about Human Goals

Knowledge about human goals has been found to be an important kind of knowledge for a range of challenging research problems, such as goal recognition from people's actions, reasoning about people's goals or the generation of action sequences that implement goals (planning) [Schank and Abelson, 1977]. In contrast to other kinds of knowledge, e.g. commonsense, knowledge about human goals provides a different perspective on textual resources putting more emphasis on future aspects and activities. We regard the acquisition of this knowledge as a first step towards conducting complex tasks such as planning.

Regardless whether the knowledge to extract is about human goals, commonsense [Liu and Singh, 2004] or the world in general [Schubert and Tong, 2003; Clarke, 2009], the acquisition process often includes the application of indication and extraction patterns. Moreover, knowledge acquisition approaches differ in how much manual intervention is necessary (or desired) in the knowledge acquisition process. Existing approaches include utilizing human knowledge engineering [Lenat, 1995], volunteer-based [Liu and Singh, 2004], game-based [Lieberman et al., 2007; von Ahn, 2006] or semiautomatic approaches [Eslick, 2006]. Yet, in this paper we are interested in approaching the question how knowledge about human goals can be automatically derived from social media text, in our case weblogs. To give an example, here is a snippet of a blog post where human goals are underlined:

> Last September, we moved into our new home. I had plans for this home, the first house--not apartment--my husband and I would live in. I was going to refinish some hand-me-down furniture we have, and I was going to plant a wonderful garden, starting with bulbs that would bloom in the spring. Crocuses, hyacinths, tulips--all of my favorites. And I would know, all winter long, that they were sleeping in the dark, cold soil, waiting to awake with the first light and warmth of spring

Though weblogs exhibit some disadvantages when it comes to quality issues, e.g., textual content is prone to noise, we can expect that weblogs contain a broad variety of human goals. In the remaining part, we describe our approach to address goal extraction from open text by deriving and evaluating a first set of lexico-syntactic patterns. We then discuss strengths and weaknesses of our patterns based on a small human subject study in order to improve them in future steps.

2 Patterns To Extract Human Goals

We employ and adapt the definition from [Tatu, 2005] who defines human goals as: "*Expressions of a particular action that shall take place in the future, in which the speaker is some sort of agent.*" The following, exemplary sentence taken from the blog post snippet presented above: "I was going to refinish some hand-me-down furniture" indicates the person's intention to prettify some furniture. In contrast to Tatu's definition, we do not include information about the speaker into our patterns to keep them simple. However, the idea to include this kind of information is discussed in Section 3.3. When comparing our setup to [Tatu, 2005]'s, we can observe three differences. Firstly, the author developed part-of-speech patterns by annotating and examining samples from the Brown corpus. Working on the Brown corpus is advantageous because this corpus has already been tagged – the chance of getting incorrect part-of-speech tags is thereby reduced. Secondly, the language used in the Brown corpus is different than language used in weblogs. Thirdly, [Tatu, 2005]'s motivation to address challenges in question answering (QA). She expected that sentences containing expressions of human goals are better suited to answer a certain kind of questions. Textual resources in the QA domain exhibit other characteristics than weblogs, for instance, people use weblogs to tell stories or write diary-like entries. We hypothesize that extraction patterns yield different results depending on weblog characteristics, e.g., does the weblog contain a story-like structure or not?

We followed a common path to acquire knowledge from textual resources by manually examining the textual environment to identify appropriate patterns [Hearst, 1992]. As a first step, we drew a small, random sample (~ 100 blog posts) from the ICWSM 2009 Spinn3r Dataset [Burton et al., 2009] and annotated the textual contents according to the above definition. The annotation task was conducted by one of the authors and an undergraduate student. Table 1 illustrates ten resulting, lexico-syntactic patterns based on these annotations which are partly inspired by patterns by [Tatu, 2005]. She employs these patterns to identify sentences containing intentional expressions in order to build up a training set for further experiments. Part-of-speech tags throughout this paper are consistent with the Penn Treebank Tag Set.

Table 1: Lexico-syntactic patterns to identify and extract human goals and matching instances. (*) denotes no, one or several occurrences, (+) denotes at least one occurrence, (?) denotes one optional occurrence and (|) denotes a logical OR.

Nr.	Lexico-Syntactic Patterns	Matching Instances
1	<VB|VBZ> <TO> <VB>	needs/VBZ to/TO organize/VB
2	<NN.*> <TO> <VB>	alcohol/NN to/TO get/VB
3	<JJ> <TO> <VB.*>	available/JJ to/TO read/VB
4	<VB> <DT> <NN.*>	find/VB a/DT keyboard/NN
5	<WANT> <TO> <VB>	wanted/VBD to/TO kill/VB
6	<INTEND> <TO> <VB>	intend/VBP to/TO quit/VB
7	<INTENT|PURPOSE|GOAL| OBJECTIVE><VBZ><TO><VB| NN.*>*	goal/NN is/VBZ to/TO eat/VB
8	<LIKE> <TO> <VB.*>	like/VB to/TO share/VB
9	<WANT> < PRP> <TO> <VB>	wants/VBZ them/PRP to/TO go/VB
10	<GET> <PRP> <DT>? <NN.*> | <VB.*>	get/VB you/PRP to/TO purchase/VB

In the next section, we apply our extraction patterns to a larger sample of weblogs. We then evaluate the quality of every pattern by calculating precision values.

3 Quality & Characteristics

In this section, we briefly describe our data preparation steps and pattern matching process. We report precision results of preliminary study on a set of ~205.000 blog posts and discuss observed weaknesses of our patterns. We conclude this section with suggesting several possibilities to improve and extend the patterns to extract knowledge about human goals.

3.1 Data Sets

For our experiments, we used the ICWSM 2009 Spinn3r Dataset which comprises 44 million blog posts made between August 1st and October 1st, 2008. We randomly drew ~205.000 blog posts and further separated them into two datasets – one with posts containing stories – one with posts containing non-stories. We hypothesize that blog posts telling a story contain more human goals than other blog posts. We use work from [Gordon and Reid, 2009] that defines a story as a series of causally related events in the past. They developed an automatic algorithm to identify blog posts most likely containing a story (reported precision values up to 75%). Moreover, they provide an index of all blog posts in the ICWSM 2009 Spinn3r Dataset that were classified as containing story-like structures. Using this information, we obtained two datasets – one containing posts with stories (~3000) and one containing posts without stories (~202.000).

3.2 Data Preparation

We first extracted the content of the <description> field in the corresponding xml files of the random sample. Since the textual content of the weblogs was often messy, we had to clean it as preparation for the subsequent part-of-speech tagging. The cleaning procedure included removing html snippets and special characters. For the process of part-of-speech tagging and pattern matching, we used functionality of the Natural Language Processing Toolkit (NLTK[1]) in combination with Python as programming language.

3.3 Strengths and Weaknesses of our Goal Extraction Patterns

We applied our patterns from Table 1 to two datasets (see Section 3.1) which were randomly drawn from the ICWSM 2009 Spinn3r Dataset (tiergroups 1-3).

Table 2 shows the number of matches per extraction pattern. The frequency numbers corroborate our hypothesis that there is a higher potential for the presence of human goals weblogs containing a story. Since there are ~67 times more blog posts containing non-stories than stories, the numbers are not directly comparable. In order to compare them, we calculate the ratio of (number of found goal instances) vs. (number of blog posts). We notice that the ratio is always highly in favor of the blog posts containing stories. Consider for example ratios for the first pattern <VB|VBZ> <TO> <VB>: 486/3,000 = 0.16 for stories vs. 6,220/202,000 = 0.03 for non-stories.

Table 2 illustrates the number of matched goal instances per extraction pattern as well as precision values (sample size of 20) for both story and non-story content.

Lexico-Syntactic Patterns	Story Set (#3.000)		Non-Story Set (#202.000)	
	Freq.	Prec.	Freq.	Prec.
<VB|VBZ> <TO> <VB>	486	0.1	6220	0
<NN.*> <TO> <VB>	2018	0	6661	0
<JJ> <TO> <VB.*>	677	0.05	5424	0.06
<VB> <DT> <NN.*>	1405	0.06	15129	0
<WANT> <TO> <VB>	398	0,53	3614	0.37
<INTEND> <TO> <VB>	10	0.6	86	0.5
<INTENT|PURPOSE|GOAL|OBJECTIVE><VBZ><TO><VB| NN.*>*	2	0.5	39	0.82
<LIKE> <TO> <VB.*>	36	0.16	592	0.18
<WANT> < PRP> <TO> <VB>	30	0.83	291	0.11
<GET> <PRP> <DT>? <NN.*> | <VB.*>	16	0.25	47	0.32

For every pattern, an undergraduate student rated a maximum number of 40 matched instances whether a human goal is expressed or not. The student took the context (sentence boundary) into account when he rated the matched instances. The precision values for every pattern are calculated based upon 20 instances from story content and 20 instances from non-story content. In five cases, where the pattern matched fewer than 20 instances, the precision values are based on a slightly lower number of rated samples.

[1] http://www.nltk.org/

To discuss strengths and weaknesses, we group our patterns into three categories and provide positively and negatively rated instances per pattern category, i.e. true positives and false positives. The first category (Nr. 1 to 4) contains pure part-of-speech patterns, the second category (Nr. 5 to 8) contains part-of-speech patterns combined with goal keywords and patterns of the third group (Nr. 9 to 10) can be expected to extract not only goal knowledge but to extract additional information on the participants involved.

We can observe that precision values in the first category are low. Though these pure part-of-speech patterns produce a lot of matches, the matched instances appear too general and are therefore inappropriate to extract human goals. Moreover, positive examples are partly matched by other categories such as "want him to learn to ride a bike" which actually serves as positive examples for patterns Nr. 1 and Nr. 4. Table 3 shows true and false positives extracted by these patterns.

Table 3 shows true and false positives of extracted human goals (from patterns Nr.1-4).

Matched Instance	Context	Goal
learn/VB to/TO ride/VB	that want him to learn to ride a bike.	yes
have/VB to/TO agree/VB	I might have to agree on some levels	no
car/NN to/TO go/VB	We got in the car to go to the hospital	no
things/NNS to/TO load/VB	I just have a few more things to load	no
willing/JJ to/TO believe/VB	I am willing to believe in love	yes
ready/JJ to/TO take/VB	I was ready to take that chance with you	no
ride/VB a/DT bike/NNP	that want him to learn to ride a bike	yes
take/VB another/DT night/NN	I can't take another night of this	no

Patterns in the second category almost all achieved a precision value higher than 50% except for two patterns. The first one is pattern Nr. 8. When reviewing the matched instances, we find sentences such as: "She swims a lot and likes to drink lake water." (see Table 4) where a person's preferences are expressed. We would rather like to match sentences such as "I like to play soccer in the evening" implying an action that takes place in the future. Therefore, in order to improve this pattern, we could require the presence of certain temporal expressions such as 'today' or 'tomorrow'. The second exception is pattern Nr. 7 (story content) where the moderate precision value is most likely due to the low number of matches. In case of the non-story content, this pattern yields high precision values demonstrating its usefulness.

Table 4 shows true and false positives of extracted human goals (from patterns Nr.5-8).

Matched Instance	Context	Goal
wanted/VBD to/TO go/VB	I never wanted to go back to school	yes
wants/VBZ to/TO do/VB	he wants to do it	no
intend/VBP to/TO get/VB	I intend to get up at 7:30	yes
intend/VB to/TO stay/VB	Jean, do you intend to stay here until morning?	no
goal/NN is/VBZ to/TO eat/VB	the ultimate goal is to eat the cookie	yes
goal/NN is/VBZ to/TO	on what makes a safe tire. With all your communications, you goal is to.	no
like/VB to/TO move/VB	We would like to move into this house sometime before the year 2020	yes
like/VBP to/TO do/VB	She swims a lot and likes to drink lake water.	no

Examining the remaining three negative examples in Table 4, we can gain further insights on how to improve the extraction patterns. The first negative instance: "He wants to do it" can be avoided by simply requiring the pattern to end in a verb phrase.

<WANT> <TO> <VB> → <WANT> <TO> (<VB><DT>?<JJ>*<NN.*>+)
<INTEND> <TO> <VB> → <INTEND> <TO> (<VB><DT>?<JJ>*<NN.*>+)

The second negative instance: "Jean, do you intend to stay here until morning?" suggests to check whether the sentence is interrogative or not. A simple approach would be to take punctuation information into account, yet in case of weblog content punctuations might not always be provided.

The third negative instance: "With all your communications, you goal is to" can be easily avoided by adapting the pattern to require at least one verb or noun after the part-of-speech tag <TO>. In the same step, one might think of including plural forms of the keywords "goal, purpose, intent and objective" into the pattern.

<GOAL><VBZ><TO><VB| NN.*>* → <GOAL><VBZ><TO><VB| NN.*>+

In summary, we can speculate that the more patterns take advantage of context information the more accurate the extracted instances are. True and false positives of patterns Nr.5-8 are illustrated in Table 4.

Patterns in the third category exhibit an additional characteristic compared to the other two categories. To give an example where pattern Nr.9 matched following chunk (wanted/VBD me/PRP to/TO buy/VB) that was part of the sentence: "he wanted me to buy him a new chair". Besides the purchase of a chair as a future action, we learn something about the relation among the participants. This information can be used when trying to identify the goal carrier, i.e. the person who actually issues her goal.

Table 5 illustrates true and false positives that were extracted by patterns Nr. 9 and Nr. 10.

Table 5 shows true and false positives of extracted human goals (from patterns Nr. 9-10).

Matched Instance	Context	Goal
wants/VBZ me/PRP to/TO send/VB	He wants me to send him a hard copy	yes
wanted/VBD it/PRP to/TO be/VB	I wanted it to be about me.	no
get/VB him/PRP to/TO email/VB	Mum went to contact her former pupil and get him to email me	yes
get/VB you/PRP to/TO do/VB	really have better things to do, so I'm going to get you to do it	no

3.4 Potential Extensions

In the previous subsection, we described various improvement strategies that were based on examining a small sample of false positives. In this subsection, we discuss potential benefits of including other feature types than only keywords and part-of-speech tags.

At present state, many false positives are due to incorrect part-of-speech tagging. We intend to examine whether we could become independent from tagging quality by, for example, only using lexical and punctuation features. A conceivable approach could be to (i) identify indicators such as “intend to” and (ii) take the remaining tokens till the next punctuation is reached. Regular expressions represent a means to implement this approach which is then to be evaluated.

A more complex approach involves identifying the verb’s agent to ensure the identified goal belongs to a person [Tatu, 2005]. Including this feature could avoid false positives such as: “The dog is going to bite the postman”. However, the annotation of open text with linguistic features such as semantic roles and predicate argument structures is challenging. Challenges include (i) incorrect part-of-speech tagging and (ii) lack of adequate tools to annotate semantic roles. Thus, we suggest employing linguistic resources such as PropositionBank [Palmer et al., 2005] and FrameNet [Baker et al., 1998] that might help to evolve our patterns. The main advantage, for example, of the PropositionBank corpus is that it is already annotated not only with part-of-speech tags and parse tree information but with predicate-argument structures as well. An interesting next step would be to apply our patterns to the PropositionBank corpus to learn how the additional, linguistic information can be included in our patterns.

4 Summing Up

In this paper, we present work in progress towards generating a knowledge base that contains a broad spectrum of human goals. By analyzing such a knowledge base of human goals, we could learn something about people’s current (crawling date) motivations and intentions. We expect that weblogs are an appropriate source for acquiring this knowledge. As a first step, we evaluate a set of simple patterns to identify human goals in weblogs. We intend to extend these patterns based on our observations, above all to include verb phrases – apparently the predominant carrier of human goal expressions. With regard to our objective of generating a knowledge base of human goals, we deem precision of our patterns more important than recall.

We reckon that aspects of this work could inform social applications e.g. search in weblogs. While traditional systems search and rank weblogs based on content information, information about the goal a blogger pursues has not yet been taken into account. One could imagine a blog search system that ranks weblogs based on shared goals. Knowledge about goals could also provide a novel way to identify weblog communities. Participants of these communities would not only share interests but also common goals.

Acknowledgments

Thanks to Daniel Lamprecht for participating in the data cleaning and annotation tasks. This work is funded by the FWF Grant P20269 *TransAgere* and the Know-Center.

References

[Baker et al., 1998] C. Baker, C. Fillmore and J. Lowe. The Berkeley FrameNet Project, in 'Proceedings of the 17th international conference on Computational linguistics', Association for Computational Linguistics, Morristown, NJ, USA, pp. 86—90, 1998.

[Burton et al., 2009] K. Burton, A. Java, and I. Soboroff. The ICWSM 2009 Spinn3r Dataset. In Proceedings of the Third Annual Conference on Weblogs and Social Media (ICWSM 2009), San Jose, CA, May 2009.

[Clarke and Harrison, 2009] P. Clark and P. Harrison. Large-Scale Extraction and Use of Knowledge From Text. In The Fifth International Conference on Knowledge Capture, 2009.

[Eslick, 2006] I. Eslick. Searching for commonsense, Master’s thesis, Massachusetts Institute of Technology, 2006.

[Gordon and Swanson, 2009] A. Gordon and R. Swanson. Identifying Personal Stories in Millions of Weblog Entries. Third International Conference on Weblogs and Social Media, Data Challenge Workshop, San Jose, CA, May 20, 2009.

[Hearst, 1992] M. Hearst. Automatic acquisition of hyponyms from large text corpora, in 'Proceedings of the 14th conference on Computational linguistics', Association for Computational Linguistics, Morristown, NJ, USA, pp. 539—545, 1992.

[Lenat, 1995] D. Lenat. CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM, (38)11:33-38, 1995.

[Lieberman et al, 2007] H. Lieberman, D. Smith, A. Teeters. Common Consensus: a web-based game for collecting commonsense goals. In Proceedings of the Workshop on Common Sense and Intelligent User Interfaces held in conjunction with the 2007 International Conference on Intelligent User Interfaces, IUI, 2007.

[Liu and Singh, 2004] H. Liu and P. Singh. ConceptNet - A practical commonsense reasoning tool-kit. BT Technology Journal, (22)4:211-226, 2004.

[Palmer et al., 2005] M. Palmer, D. Gildea and P. Kingsbury. The Proposition Bank: A Corpus Annotated with Semantic Roles, Computational Linguistics Journal, 2005.

[Schank and Abelson, 1977] R. Schank and R. Abelson R: Scripts, plans, goals, and understanding: an inquiry into human knowledge structures. Lawrence Erlbaum Associates, 1977.

[Schubert and Tong, 2003] L. Schubert and M. Tong. Extracting and evaluating general world knowledge from the Brown corpus, in 'Proceedings of the HLT-NAACL 2003 workshop on Text meaning', Association for Computational Linguistics, Morristown, NJ, USA, pp. 7—13, 2003.

[Singh et al., 2002] P. Singh, T. Lin, E. Mueller, G. Lim, T. Perkins and W. Zhu. Open Mind Common Sense: Knowledge acquisition from the general public. In Proceedings of the First International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems, 1223--1237, Springer-Verlag London, UK, 2002.

[Tatu, 2005] M. Tatu. Automatic Discovery of Goals in Text and its Application to Question Answering. In 43rd Annual Meeting of the Association for Computational Linguistics, 'ACL 05’, 2005.

[von Ahn, 2006] L. von Ahn. 'Games with a Purpose', Computer 39(6), 92—94, 2006.

Key Expression driven Record Mining for Event Calendar Search

Yeong Su Lee & Michaela Geierhos
Centrum für Informations- und Sprachverarbeitung
Ludwig-Maximilians-Universität München
{yeong|micha}@cis.uni-muenchen.de

Abstract

This paper presents an approach to extract data records from websites, particularly ones with event calendars. We therefore use language-specific key expressions and HTML patterns to recognize every single event given on the investigated web page. One of the most remarkable advantages of our method is that it does not require any additional classification steps based on machine learning algorithms or keyword extraction methods; it is a so-called one-step mining technique. Our experimental results obtained on German opera websites show excellent results in precision and recall. Furthermore, we could demonstrate that our proposed technique outperforms other data record mining applications run on event sites.

1 Introduction

There are numerous web sites providing large databases containing information such as yellow page listings or event calendars. A user typically accesses such a database over the Internet by using a web browser. The requested information is to be displayed in dynamically generated web pages by a script using a back-end database. During the last years, several parsing algorithms have been developed to automatically determine the information record boundaries and extract the corresponding data records

Current approaches to data record mining [Arasu and Garcia-Molina, 2003; Crescenzi *et al.*, 2001; Lerman *et al.*, 2003] exploit the structured character of HTML documents. For this purpose, two ore more similar web pages have to be compared in order to extract the corresponding data records. These systems often expect preclassified web pages as input (cf. RoadRunner [Crescenzi *et al.*, 2001]). Based on the fact that data records are dynamically generated from a back-end database, some applications like MDR [Liu *et al.*, 2003] try to reconstruct the given web page benefiting from the regularities of HTML structure. Hereby, the main focus is to determine iterations of HTML tag sequences by using the DOM tree representation of a web page. Since most structured web data is arranged in rows and columns, many researchers have concentrated on table recognition for information extraction purposes [David W. Embley, 2006]. But the recent shift away from the HTML tag $\langle table \rangle$ to both cascading style sheets (CSS) and the $\langle div \rangle$ tag complicates the identification of data records.

Due to the loose strictness of HTML, others try to exploit the visual information provided by a web browser. Like [Cai *et al.*, 2003] many researchers [Algur and Hiremath, 2006; Hiremath *et al.*, 2005; Liu *et al.*, 2006; Zhao *et al.*, 2005] have proposed to use rendering techniques for data record mining. They apply these methods on the displayed query results in order to determine the record boundaries. Even flat and nested data records can be extracted by the VSAP [1] technique [Hiremath *et al.*, 2005] based on some heuristics [Algur and Hiremath, 2006]. Although rendering methods achieve good results, they have one big drawback: They require a web browser, that correctly displays the investigated web page, to determine some typical visual cues of a data region (e.g. size, background color, icons, font colors).

Regardless of the technique used, all methods have one point in common: Current approaches to data record mining disregard any language-specific information. We observed that existing data record mining techniques are not satisfactory enough for specialized search purposes limited to restricted domains. The success of event calendar search highly depends on language-specific trigger words indicating at least some date.

We therefore propose a novel and robust method for data record mining on demand. Browsing restricted domains allows us to define some key words, e.g. weekdays, nested in the data records of event sites. By means of limited vocabulary, we are able to analyze the document's HTML structure and locate the corresponding data record boundaries. Our technique is quite robust in variability of the DOM, upgradeable and keeps data up-to-date. As our case study was limited to German opera websites, the investigated language was German. However, the developed technique is adaptive to non-German websites with slight language-specific modifications, and experimental results on real-life websites confirm the feasibility of the approach[2].

The paper is structured as follows. In the next section we introduce the concepts and terms used in the paper. In section 3, we present our data mining technique. Section 4 evaluates the proposed method. In section 5, we summarize our work and finally highlight future research directions.

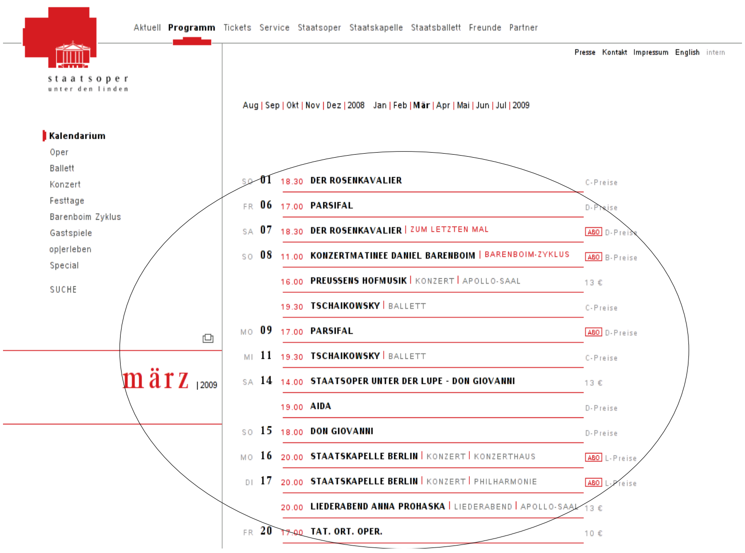
2 Definition of terms

Terms that are used throughout this paper in a particular usage have to be clearly defined.

A large amount of information on the Web is contained in regularly structured objects, which we call data records.

[1] Visual Structure based Analysis of web Pages

[2] For research and test purposes the prototype of our system is available at `http://www.cis.uni-muenchen.de/~yeong/EVENTSEARCH/eventsearch.html`.



SO	01	18.30	DER ROSENKAVALIER	C-Preise
FR	06	17.00	PARSIFAL	D-Preise
SA	07	18.30	DER ROSENKAVALIER | ZUM LETZTEN MAL	ABO D-Preise
SO	08	11.00	KONZERTMATINEE DANIEL BARENBOIM | BARENBOIM-ZYKLUS	ABO B-Preise
		16.00	PREUSSENS HOFMUSIK | KONZERT | APOLLO-SAAL	13 €
		19.30	TSCHAIKOWSKY | BALLETT	C-Preise
MO	09	17.00	PARSIFAL	ABO D-Preise
MI	11	19.30	TSCHAIKOWSKY | BALLETT	C-Preise
SA	14	14.00	STAATSOPER UNTER DER LUPE - DON GIOVANNI	13 €
		19.00	AIDA	D-Preise
SO	15	18.00	DON GIOVANNI	D-Preise
MO	16	20.00	STAATSKAPELLE BERLIN | KONZERT | KONZERTHAUS	ABO L-Preise
DI	17	20.00	STAATSKAPELLE BERLIN | KONZERT | PHILHARMONIE	ABO L-Preise
		20.00	LIEDERABEND ANNA PROHASKA | LIEDERABEND | APOLLO-SAAL	13 €
FR	20	17.00	TAT. ORT. OPER.	10 €

Figure 1: Sample web page with event calendar records

Such data records (...) often present the essential information of their host pages [Liu *et al.*, 2003], e.g. event calendars.

Definition 1 (Event calendar record)
An event calendar record is primarily a data record which provides information on event details like event title, event location, event date, etc.

In order to distinguish event calendar records from ordinary data records hosting, for example, the search menu for event style, we have to define some vocabulary describing an event, so-called key expressions.

Definition 2 (Key Expression)
An instance of a feature set that classifies a data record and can be described by regular expressions or string variants is called key expression.

Every event calendar record contains useful information including weekday, date, time, title and price (cf. Figure 1). Among these, we can easily identify date, time and weekday by using some regular expression and a list of weekdays together with their corresponding abbreviations.

```
date = [0-9]{2}\W[0-9]{2}\W[0-9]{4}\W
wdAbbrev = (Mo|Di|Mi|Do|Fr|Sa|So)
wd = ((Mon|Diens|Donners|Frei|Sams|Sonn)tag
       |Mittwoch|Sonnabend)
```

Other attributes like event title and price, and other additional information can be either difficult to recognize (e.g. title) or optional (e.g. price). Thus, we use date, time and weekday as key terms (a seed list) to search for an event calendar record. Having detected such a record, we extract all information bits found for the corresponding event.

Please note that the selection of a key expression highly depends on the record type. For example, within a shopping record, the key expression may be some price information, and within a computer description, it may be the CPU type.

3 The proposed technique

After retrieving a large website, each web page has to be classified into pages with event calendars or without, depending on its key expressions. If the page contains at least two or more key expressions, then the search for event calendar records will start. Otherwise, the page will be skipped. Thus, the classification of pages containing event calendar records can be performed without the help of machine learning algorithms or keyword extraction methods.

We now present the two steps of our approach:

1. First, we create the DOM tree of a selected web page in order to exploit its HTML structure (cf. section 3.1)
2. Secondly, we assume that there is only one smallest maximum data region for the event calendar records [Liu *et al.*, 2006; Hiremath *et al.*, 2005; Liu *et al.*, 2003; Zhai and Liu, 2005] and it corresponds to only one HTML tag region. The smallest maximum data region can be determined by a top-down traversal of the tree using key expressions (cf. section 3.2). It is predictable that there must be two or more key expressions in one HTML tag region of the tree. Otherwise, this tag region will be cut off from the DOM tree.

3.1 Exploiting the structure of event calendar records within the DOM tree

Each website has its own distinct method of presenting information. Therefore, the high variability observed in HTML structure should be taken into account. However, the number of possible tag combinations which can be considered for event calendar records is very limited.

The following types of event calendar records according to their tree structure have been registered and were classified as follows:

1. One single record under one node
2. All records under one node – each record consists of a set of children nodes (cf. Figure 2)

In the first case, we act on the assumption that each data record represented by one HTML tag region has no siblings. If this tag region contains some key expression (e.g. weekday) and other event-related information, it will be selected as event calendar record.

In contrast to the first type, all event calendar records are siblings (cf. Figure 2) belonging to the same parent node.

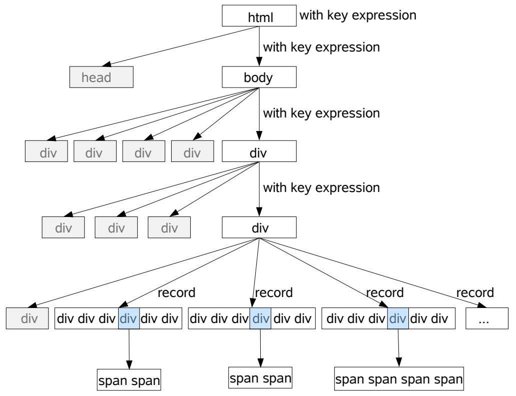


Figure 2: The core tree structure of Figure 1

We can thereby distinguish between three structure types of data records (HTML tag regions) depending on the co-occurrence of tag attributes and values.

(a) repetition of an HTML tag, e.g. ⟨div⟩, with non-recurring attributes across all data records, including their text values,

(b) repetition of an HTML tag, e.g. ⟨div⟩, with recurring attributes across all data records and sometimes incomplete attribute-value-pairs (cf. Figure 3),

(c) missing both tag attribute and value.

Among these, case (a) is really rare and (c) can sometimes happen, but in practice, case (b) occurs quite frequently.

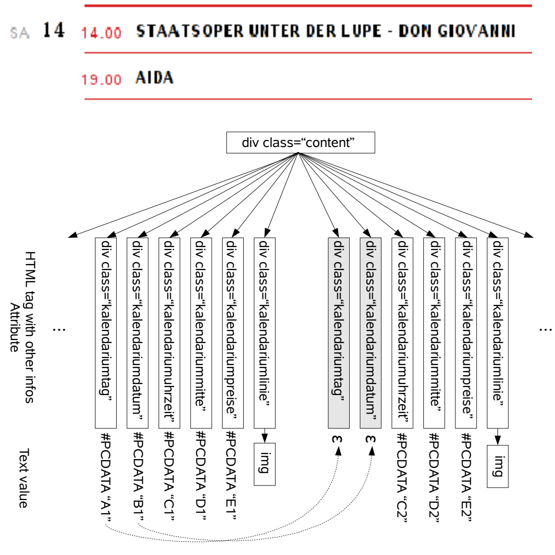

SA 14 14.00 STAATSOPER UNTER DER LUPE - DON GIOVANNI
19.00 AIDA
div class="content"
HTML tag with other infos Attribute
Text value
...
div class="kalendariumtag"
div class="kalendariumdatum"
div class="kalendariumuhrzeit"
div class="kalendariummitte"
div class="kalendariumpreise"
div class="kalendariumlinie"
div class="kalendariumtag"
div class="kalendariumdatum"
div class="kalendariumuhrzeit"
div class="kalendariummitte"
div class="kalendariumpreise"
div class="kalendariumlinie"
...
#PCDATA "A1"
#PCDATA "B1"
#PCDATA "C1"
#PCDATA "D1"
#PCDATA "E1"
img
ε
ε
#PCDATA "C2"
#PCDATA "D2"
#PCDATA "E2"
img

Figure 3: Tag iteration using various attributes and missing text values within an event calendar record (b)

In Figure 2, we showed that the key expression is inherited to only one HTML tag node (⟨html⟩ → ⟨body⟩ → the 5^{th} ⟨div⟩ → the 4^{th} ⟨div⟩), and all records are the children of this one single node (cf. ⟨div class=“content”⟩ in Figure 3). When we zoom in and look at the record structure in detail, each record is composed of six ⟨div⟩ tags and its corresponding attributes: *“kalendariumtag”*, *“kalendariumdatum”*, *“kalendariumuhrzeit”*, *“kalendariummitte”*, *“kalendariumpreise”* and *“kalendariumlinie”*. As shown in Figure 3, the text values (ε) are missing for the first two mentioned HTML tag attributes in the one record, but are filled with #PCDATA “A1” and #PDCATA “B1” in the preceding record. This means that the text values of the first record are also valid for the following record. In our case, the opera *Don Giovanni* takes place on the same day (kalendariumtag=Sa, kalendariumdatum=14) as *Aida* (kalendariumtag=ε, kalendariumdatum=ε). We therefore resolve such co-references by linking the text values of the same attributes in successive records.

Assuming that the two tag attributes displayed in gray are also missing, Figure 3 would be an example for (c).

But one problem that still remains to be solved is how to decide where the record starts and where it ends: The boundary between records can be determined by comparing the bordering tag attributes. Based on the assumption that the key expression, e.g. weekday, is placed in first position (cf. #PCDATA “A1” in Figure 3), then we have two possibilities: We can go forward or backward to recognize the record boundaries. If we move forward, the same tag attribute will recur after six steps. That way, we learn that one record consists of six tag attributes. However, we do not know yet where the record begins. In order to solve this problem, we go back until we find a tag attribute totally different from the six common attributes in Figure 3. Now we can initialize the starting points for all records embracing six tag attributes each.

In all cases, we try to correctly determine the smallest maximum data region.

3.2 Decision of smallest maximum data region

After classifying an HTML document as event calendar page, we have to detect the smallest maximum data region containing some key expression within the DOM tree. Assuming that the smallest maximum data region only consists of event calendar records, every record must have its own parent node.

```
sub scanSmallestMaxDataRegion {
 node = shift;
 for each child (node->contentList)
  if (child has a set of nodes)
    if (child->asHTML() matches keyExp)
      then smallestMaxDataNode = child;
      scanSmallestMaxDataRegion(child);
    else
      scanSmallestMaxDataRegion(child);
    endif
  else
    return 0;
  endif
 endfor
 return smallestMaxDataNode;
}
```

Figure 4: Pseudo-algorithm for decision process of the smallest maximum data region

In Figure 4 is described how to determine the smallest maximum data region. In short, we launch a top-down-traversal of the DOM tree starting at the root node. We then search the content of the child nodes for key expressions in order to detect the regularities in their occurrences. That way, we measure the maximum distance of a re-appearing type of key expression, e.g. weekday, and determine the corresponding subtree within the HTML structure. Our understanding of *data region* is thereby the same as in [Hiremath *et al.*, 2005]. Once, the smallest maximum data region is located, each record must be mined by using some key expression.

4 Experimental evaluation

To evaluate the quality of the proposed record mining technique from arbitrary websites, we concentrate our case study on websites of German opera houses. Our test set consists of eleven event calendar pages randomly retrieved from websites of opera houses dated on April 14, 2009.

Table 1: Evaluation results

URL	Record Objects	KEDR[3]	Recall
bayerische.staatsoper.de	21	20	95.24%
staatsoper-berlin.org	33	33	100.00%
theater-chemnitz.de	10	10	100.00%
oper-frankfurt.de	26	26	100.00%
oper-halle.de	24	24	100.00%
hamburgische-staatsoper.de	17	17	100.00%
oper-hannover.de	26	26	100.00%
oper-leipzig.de	27	27	100.00%
rheinoper.de	12	12	100.00%
semperoper.de	37	37	100.00%
staatstheater.stuttgart.de	37	37	100.00%
On average			99.56%

Needless to say, the evaluation results displayed in Table 1 show excellent results of recall (99.56%) and we constantly obtained a precision of 100%. The reason for this lack of recall is due to the non-recognition of the time attribute within a data record. As mentioned in section 2, every event calendar record contains weekday, date, time, title and price information. For a full recognition of a record, at least weekday and time have to be correctly determined. Assuming that one aspect is missing, we count this record as false negative. For lack of specification, our algorithm could not identify the time for one stage performance in April's event calendar of Bayerische Staatsoper (95.24%).

One presumption is that our key expression driven record mining technique expects a valid HTML page for the DOM tree construction. If the tree cannot be built up, we will use some open source tools, e.g. *tidy*[4], for correction purposes. We must admit that our approach is not able to reconstruct the DOM tree of web pages with no closing HTML tags. Running other comparable data record mining applications like MDR [Liu *et al.*, 2003] on our test web pages does not produce any noteworthy results. The event calendar records cannot be either located or correctly assigned to the corresponding data regions. Therefore, recall and precision are vanishing small.

5 Conclusion and future work

Since current approaches to data record mining have disregarded any language-specific information and only exploited the structured character of HTML, we combine both: HTML patterns with predefined key expressions.

Our future work will concentrate on both automated key expression learning and substructure analysis of data records. By measuring the similarity of content strings or tag regions, we will figure out the best candidates for domain-specific key expressions. Moreover, it seems essential to validate this approach on a much larger test set demonstrating that web pages of that type show little variability. Besides, it could be interesting to test this technique on other websites with event information (e.g. sports).

[3] Key Expression Driven Records

[4] `http://tidy.sourceforge.net`

References

[Algur and Hiremath, 2006] Siddu P Algur and P S Hiremath. Visual Clue Based Extraction of Web Data from Flat and Nested Data Records. In *International Conference on Management of Data (COMAD 2006)*, Dehli, India, 2006.

[Arasu and Garcia-Molina, 2003] Arvint Arasu and Hector Garcia-Molina. Extracting Structured Data from Web Pages. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 337–348, San Diego, California, USA, 2003.

[Cai *et al.*, 2003] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. Extracting Content Structure for Web Pages based on Visual Representation. In *Web Technologies and Applications: 5th Asia-Pacific Web Conference, APWeb 2003*, Xian, China, 2003.

[Crescenzi *et al.*, 2001] Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo. RoadRunner: Towards Automatic Data Extraction from Large Web Sites. In *Proceedings of the 27th VLDM Conference*, Rome, Italy, 2001.

[David W. Embley, 2006] George Nagy David W. Embley, Daniel Lopresti. Notes on contemporary table recognition. In *Proceedings. Document Analysis Systems VII, 7th International Workshop, DAS 2006*, volume 3872, pages 164–175. Springer, Berlin, Germany, 2006.

[Hiremath *et al.*, 2005] P S Hiremath, S S Benchalli, Siddu P Algur, and Renuka V Udapudi. Mining Data Regions from Web Pages. In *International Conference on Management of Data (COMAD 2005)*, Hyderabad, India, 2005.

[Lerman *et al.*, 2003] Kristina Lerman, Lise Getoor, Steven Minton, and Craig Knoblock. Using the structure of Web sites for automatic segmentation of tables. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 119–130, Paris, France, 2003.

[Liu *et al.*, 2003] Bing Liu, Robert Grossman, and Yanhong Zhai. Mining data records in web pages. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining*, pages 601–606, Washington D.C., USA, 2003.

[Liu *et al.*, 2006] Wei Liu, Xiaofeng Meng, and Weiyi Meng. Vision–based Web Data Records Extraction. In *Ninth International Workshop on the Web and Databases (WebDB 2006)*, pages 20–25, Chicago, USA, 2006.

[Zhai and Liu, 2005] Yanhong Zhai and Bing Liu. Web Data Extraction Based on Partial Tree Alignment. In *WWW2005*, Chiba, Japan, 2005.

[Zhao *et al.*, 2005] Hongkun Zhao, Weiyi Meng, Zonghuan Wu, Vijay Raghavan, and Clement Yu. Fully Automatic Wrapper Generation For Search Engines. In *WWW2005*, Chiba, Japan, 2005.

Efficient Voting Prediction for Pairwise Multilabel Classification (resubmission)[*]

Eneldo Loza Mencía, Sang-Hyeun Park and Johannes Fürnkranz
TU-Darmstadt - Knowledge Engineering Group
Hochschulstr. 10 - Darmstadt - Germany

Abstract

The pairwise approach to multilabel classification reduces the problem to learning and aggregating preference predictions among the possible labels. A key problem is the need to query a quadratic number of preferences for making a prediction. To solve this problem, we extend the recently proposed *QWeighted* algorithm for efficient pairwise multiclass voting to the multilabel setting, and evaluate the adapted algorithm on several real-world datasets. We achieve an average-case reduction of classifier evaluations from n^2 to $n+dn\log n$, where n is the total number of labels and d is the average number of labels, which is typically quite small in real-world datasets.

1 Introduction

Multilabel classification refers to the task of learning a function that maps instances $\bar{x} \in \mathcal{X}$ to label subsets $R_{\bar{x}} \subset \mathcal{L}$, where $\mathcal{L} = \{\lambda_1, \ldots, \lambda_n\}$ is a finite set of predefined labels, typically with a small to moderate number of alternatives. Thus, in contrast to multiclass learning, alternatives are not assumed to be mutually exclusive, such that multiple labels may be associated with a single instance. The predominant approach to multilabel classification is *binary relevance learning* (BR), where one classifier is learned for each class, in contrast to pairwise learning, where one classifier is learned for each pair of classes.

While it has been shown that the complexity for training an ensemble of pairwise classifiers is comparable to the complexity of training a BR ensemble [Fürnkranz, 2002; Loza Mencía and Fürnkranz, 2008b], it remained the problem that a quadratic number of classifiers has to be evaluated to produce a prediction. Our first attempts in efficient multilabel pairwise classification lead to the algorithm MLPP, which uses the fast perceptron algorithm as base classifier. With this algorithm, we successfully tackled the large Reuters-RCV1 text classification benchmark, despite the quadratic number of base classifiers [Loza Mencía and Fürnkranz, 2008b]. Although we were able to beat the competing fast MMP algorithm [Crammer and Singer, 2003] in terms of ranking performance and were competitive in training time, the costs for testing were not satisfactory. Park and Fürnkranz [2007] recently introduced a method named *QWeighted* for multiclass problems that intelligently selects only the base classifiers that are actually necessary to predict the top class. This reduced the evaluations needed from $n(n-1)/2$ to only $n\log n$ in practice, which is near the n evaluations processed by BR.

In this paper we introduce a novel algorithm which adapts the *QWeighted* method to the MLPP algorithm. In a nutshell, the adaption works as follows: instead of stopping when the top class is determined, we repeatedly apply *QWeighted* to the remaining classes until the final label set is predicted. In order to determine at which position to stop, we use the calibrated label ranking technique [Fürnkranz *et al.*, 2008], which introduces an artificial label for indicating the boundary between relevant and irrelevant classes. We evaluated this technique on a selection of multilabel datasets that vary in terms of problem domain, number of classes and label density. The results demonstrate that our modification allows the pairwise technique to process such data in comparable time to the one-per-class approaches while producing more accurate predictions.

2 Multilabel Pairwise Perceptrons

In the pairwise binarization method, one classifier is trained for each pair of classes, i.e., a problem with n different classes is decomposed into $\frac{n(n-1)}{2}$ smaller subproblems. For each pair of classes (λ_u, λ_v), only examples belonging to either λ_u or λ_v are used to train the corresponding classifier $o_{u,v}$. In the multilabel case, an example is added to the training set for classifier $o_{u,v}$ if λ_u is a relevant class and λ_v is an irrelevant class or vice versa, i.e., $(\lambda_u, \lambda_v) \in R \times I \cup I \times R$ with $I = \mathcal{L} \backslash R$ as negative labelset. The pairwise binarization method is often regarded as superior to binary relevance because it profits from simpler decision boundaries in the subproblems [Fürnkranz, 2002; Hsu and Lin, 2002; Loza Mencía and Fürnkranz, 2008b]. This allows us to use the simple but fast (linear, one-layer) perceptron algorithm as a base classifier, so that we denote the algorithm as *Multilabel Pairwise Perceptrons (MLPP)* [Loza Mencía and Fürnkranz, 2008b]. The predictions of the base classifiers $o_{u,v}$ may then be interpreted as *preference statements* that predict for a given example which of the two labels λ_u or λ_v is preferred. In order to convert these binary preferences into a class ranking, we use a simple voting strategy known as *max-wins*, which interprets each binary preference as a vote for the preferred class. Classes are then ranked according to the number of received votes. Ties are broken randomly in our case.

To convert the resulting ranking of labels into a multilabel prediction, we use the *calibrated label ranking* approach [Fürnkranz *et al.*, 2008]. This technique avoids the

[*] This manuscript is a resubmission and was already published in the Proceedings of the 17^{th} European Symposium on Artificial Neural Networks, Bruges, 22 – 24 April 2009.

```
Require: example $\bar{x}$; classifiers $\{o_{u,v} \mid u < v, \lambda_u, \lambda_v \in \mathcal{L}\}$; $l_0, \ldots, l_n = 0$
while $\lambda_{top}$ not determined do
    $\lambda_a \leftarrow \mathrm{argmin}_{\lambda_i \in \mathcal{L}}\, l_i$                                   ▷ select top candidate class
    $\lambda_b \leftarrow \mathrm{argmin}_{\lambda_j \in \mathcal{L} \setminus \{\lambda_a\}}\, l_i$ and $o_{a,b}$ not yet evaluated    ▷ select second
    if no $\lambda_b$ exists then
        $\lambda_{top} \leftarrow \lambda_a$                                             ▷ top rank class determined
    else                                                              ▷ evaluate classifier
        $v_{ab} \leftarrow o_{a,b}(\bar{x})$                     ▷ one vote for $\lambda_a$ $(v_{ab} = 1)$ or $\lambda_b$ $(v_{ab} = 0)$
        $l_a \leftarrow l_a + (1 - v_{ab})$                                    ▷ update voting loss for $\lambda_a$
        $l_b \leftarrow l_b + v_{ab}$                                          ▷ update voting loss for $\lambda_b$
```

Figure 1: Pseudocode of the *QWeighted* algorithm (multiclass classification).

dataset	n	#instances	# attributes	∅ label-set size d	density $\frac{d}{n}$	distinct
scene	6	2407	86732	1.074	17.9 %	15
yeast	14	2417	10712	4.237	30.3 %	198
r21578	120	11367	10000	1.258	1.0 %	533
rcv1-v2	101	804414	25000	2.880	2.9 %	1028
eurlex_sj	201	19596	5000	2.210	1.1 %	2540
eurlex_dc	412	19596	5000	1.292	0.3 %	1648

Table 1: Statistics of datasets.

need for learning a threshold function for separating relevant from irrelevant labels, which is often performed as a post-processing phase after computing a ranking of all possible classes. The key idea is to introduce an artificial *calibration label* λ_0, which represents the split-point between relevant and irrelevant labels. Thus, it is assumed to be preferred over all irrelevant labels, but all relevant labels are preferred over λ_0. As it turns out, the resulting n additional binary classifiers $\{\, o_{i,0} \mid i = 1 \ldots n\}$ are identical to the classifiers that are trained by the binary relevance approach. Thus, each classifier $o_{i,0}$ is trained in a one-against-all fashion by using the whole dataset with $\{\, \bar{x} \mid \lambda_i \in R_{\bar{x}}\} \subseteq \mathcal{X}$ as positive examples and $\{\, \bar{x} \mid \lambda_i \in I_{\bar{x}}\} \subseteq \mathcal{X}$ as negative examples. At prediction time, we will thus get a ranking over $n+1$ labels (the n original labels plus the calibration label). We denote the MLPP algorithm adapted in order to support the calibration technique as CMLPP.

3 Quick Weighted Voting for Multilabel Classification

As already mentioned, the quadratic number of base classifiers does not seem to be a serious drawback for training MLPP and also CMLPP. However, at prediction time it is still necessary to evaluate a quadratic number of base classifiers.

QWeighted algorithm: For the multiclass case, the simple but effective voting strategy can be performed efficiently with the Quick Weighted Voting algorithm (*QWeighted*), which is shown in Figure 1 [Park and Fürnkranz, 2007]. This algorithm computes the class with the highest accumulated voting mass without evaluating all pairwise perceptrons. It exploits the fact that during a voting procedure some classes can be excluded from the set of possible top rank classes early on, because even if they reach the maximal voting mass in the remaining evaluations they can no longer exceed the current maximum. For example, if class λ_a has received more than $n-j$ votes and class λ_b has lost j binary votings, it is impossible for λ_b to achieve a higher total voting mass than λ_a. Thus further evaluations with λ_b can be safely ignored for the comparison of these two classes. Pairwise classifiers will be selected depending on a *voting loss* value, which is the number of votes that a class has *not* received. More precisely, the voting loss l_i of a class λ_i is defined as $l_i := p_i - v_i$, where p_i is the number of evaluated incident classifiers of λ_i and v_i is the current number of votes for λ_i. Obviously, the voting loss starts with a value of zero and increases monotonically with the number of performed preference evaluations. The class with the current minimal loss is the top candidate for the top rank class. If all preferences involving this class have been evaluated (and it still has the lowest loss), we can conclude that no other class can achieve a better ranking. Thus, the *QWeighted* algorithm always focuses on classes with low voting loss.

QCMLPP1 algorithm: A simple adaptation of *QWeighted* to multilabel classification is to repeat the process. We can compute the top class λ_{top} using *QWeighted*, remove this class from $\mathcal{L}$ and repeat this step, until the returned class is the artificial label λ_0, which means that all remaining classes will be considered to be irrelevant. Of course, the information about which pairwise perceptrons have been evaluated and their results are carried through the iterations so that no pairwise perceptron is evaluated more than once. As we have to repeat this process until λ_0 is ranked as the top label, we know that the number of votes for the artificial label has to be computed at some point. So, in hope for a better starting distribution of votes, all incident classifiers $o_{i,0}$ respectively $\bar{w}_{i,0}$ of the artificial label are evaluated explicitly before iterating *QWeighted.*

QCMLPP2 algorithm: However, QCMLPP1 still performs unnecessary computations, because it neglects the fact that for multilabel classification the information that a particular class is ranked *above* the calibrated label is sufficient, and we do not need to know *by which amount*. Thus, we can further improve the algorithm by predicting the current top ranked class λ_t as relevant as soon as it has accumulated more votes than λ_0. The class λ_t is then not removed from the set of labels (as in QCMLPP1), because its incident classifiers $o_{t,j}$ may be still be needed for computing the votes for other classes. However, it can henceforth no longer be selected as a new top rank candidate.

Complexity: It is easy to see that the number of base classifier evaluations for the multilabel adaptations of *QWeighted* is bounded from above by $n + d \cdot C_{\mathrm{QW}}$,

dataset	n	BR	CMLPP	QCMLPP1	QCMLPP2	$n \log n$	$n + dn \log n$
scene	6	6	21	11.51 *(54.8%)*	11.46 *(54.6%)*	10.75	17.50
yeast	14	14	105	67.57 *(64.4%)*	64.99 *(61.9%)*	36.94	170.65
rcv1-v2	103	103	5356	485.23 *(9.06%)*	456.23 *(8.52%)*	477.38	1649.70
r21578	120	120	7260	378.45 *(5.21%)*	325.94 *(4.49%)*	574.50	843.87
eurlex_sj	201	201	20301	1144.2 *(5.64%)*	825.07 *(4.06%)*	1065.96	2556.78
eurlex_dc	412	412	85078	2610.76 *(3.07%)*	1288.22 *(1.51%)*	2480.66	3612.05

Table 2: Computational costs at prediction in average number of predictions per instance. The italic values next to the two multilabel adaptations of *QWeighted* show the ratio of predictions to CMLPP.

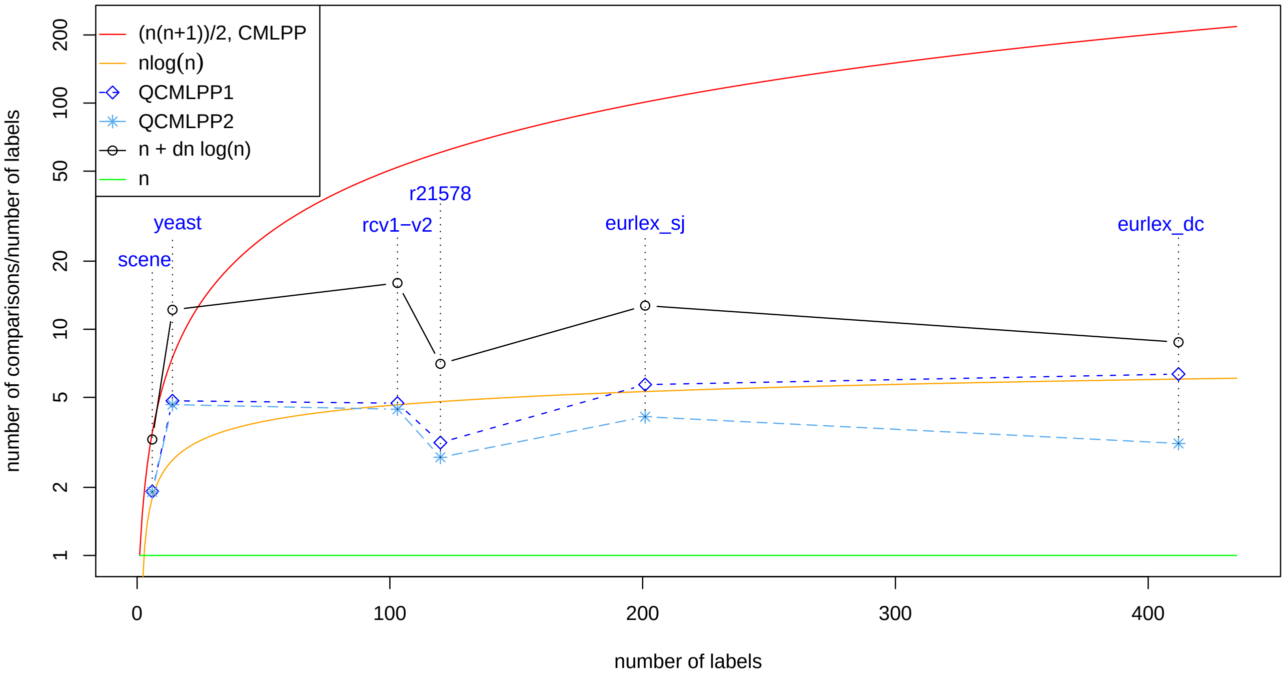


Figure 2: Prediction complexity of QCMLPP: number of comparisons needed in dependency of the number of classes n for different multilabel problems.

since we always evaluate the n classifiers involving the calibrated class, and have to do one iteration of *QWeighted* for each of the (on average) d relevant labels. Assuming that *QWeighted* on average needs $C_{QW} = n \log n$ base classifier evaluations as suggested in [Park and Fürnkranz, 2007], we can expect an average number of $n + dn \log n$ classifier evaluations for the QCMLPP variants, as compared to the $\approx n^2$ evaluations for the regular CMLPP [Fürnkranz *et al.*, 2008]. Thus, the effectiveness of the adaption to the multilabel case crucially depends on the average number d of relevant labels. We can expect a high reduction of pairwise comparisons if d is small compared to n, which holds for most real-world multilabel datasets.

4 Evaluation

Table 1 shows the multilabel datasets we used for our experiments. As the QCMLPP algorithms do not change the predictions of the CMLPP algorithm, and the superiority of the latter has already been established in other publications [Loza Mencía and Fürnkranz, 2008a,b; Fürnkranz *et al.*, 2008], we will here focus only on the computational costs. Descriptions of these datasets and results on the predictive performance may also be found in the long version of this paper [Loza Menía *et al.*, 2008]. Table 2 depicts the gained reduction of prediction complexity in terms of the average number of base classifier evaluations. In addition, we also report the ratios of classifier evaluations for the two QCMLPP variants over the CMLPP algorithm.

We can observe a clear improvement when using the *QWeighted* approach. Except for the *scene* and *yeast* datasets, both variants of the QCMLPP use less than a tenth of the classifier evaluations for CMLPP. We also add the values of $n \log n$ and $n + dn \log n$ for the corresponding datasets, which allow us to confirm that the number of classifier evaluations is smaller than the previously estimated upper bound of $n + dn \log n$ for all considered datasets. Figure 2 visualizes the above results and allows again a comparison to different complexity values such as n, $n \log n$ and n^2. Though the figure may indicate that a reduction of classifier evaluations to $n \log n$ is still achievable for multilabel classification, especially for QCMLPP2, we interpret the results more cautiously and only conclude that $n + dn \log n$ can be expected in practice.

5 Conclusions

The main disadvantage of the pairwise approach in multilabel classification was, until now, the quadratic number of base classifiers needed and hence the increased computational costs for computing the label ranking that is used for partitioning the labels in relevant and irrelevant labels. The presented QCMLPP approach is able to significantly reduce these costs by stopping the computation of the label ranking when the bipartite separation is already determined. Though not analytically proven, our empirical results show that the number of base classifier evaluations is bounded from above by $n + dn \log n$, in comparison to the evaluation of n in the case of binary relevance ranking and n^2 for the unmodified pairwise approach.

The key remaining bottleneck is that we still need to store a quadratic number of base classifiers, because each of them may be relevant for some example. We are currently investigating alternative voting schemes that use a static allocation of base classifiers, so that some of them are not needed at all. In contrast to the approach presented here, such algorithms may only approximate the label that is predicted by the regular pairwise classifier.

Acknowledgements

This work was supported by the EC 6th framework project *ALIS (Automated Legal Information System)* and by the *German Science Foundation (DFG)*.

References

Koby Crammer and Yoram Singer. A Family of Additive Online Algorithms for Category Ranking. *Journal of Machine Learning Research*, 3(6):1025–1058, 2003.

Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, 2008.

Johannes Fürnkranz. Round Robin Classification. *Journal of Machine Learning Research*, 2:721–747, 2002.

Chih-Wei Hsu and Chih-Jen Lin. A Comparison of Methods for Multi-class Support Vector Machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

Eneldo Loza Mencía and Johannes Fürnkranz. Efficient pairwise multilabel classification for large-scale problems in the legal domain. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Disocvery in Databases (ECML-PKDD-2008), Part II*, pages 50–65, Antwerp, Belgium, 2008.

Eneldo Loza Mencía and Johannes Fürnkranz. Pairwise learning of multilabel classifications with perceptrons. In *Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IJCNN 08)*, pages 2900–2907, Hong Kong, 2008.

Eneldo Loza Menía, Sang-Hyeun Park, and Johannes Fürnkranz. Advances in efficient pairwise multilabel classification. Technical Report TUD-KE-2008-06, TU Darmstadt, Knowledge Engineering Group, 2008.

Sang-Hyeun Park and Johannes Fürnkranz. Efficient pairwise classification. In *Proceedings of 18th European Conference on Machine Learning (ECML-07)*, pages 658–665, Warsaw, Poland, 2007.

Player Modeling for Intelligent Difficulty Adjustment (Resubmission)[*]

Olana Missura and **Thomas Gärtner**
Fraunhofer Institute Intelligent Analysis and Information Systems IAIS
Schloss Birlinghoven, D-53754 Sankt Augustin, Germany
firstname.lastname@iais.fraunhofer.de

Abstract

In this paper we aim at automatically adjusting the difficulty of computer games by clustering players into different types and supervised prediction of the type from short traces of gameplay. An important ingredient of video games is to challenge players by providing them with tasks of appropriate and increasing difficulty. How this difficulty should be chosen and increase over time strongly depends on the ability, experience, perception and learning curve of each individual player. It is a subjective parameter that is very difficult to set. Wrong choices can easily lead to players stopping to play the game as they get bored (if underburdened) or frustrated (if overburdened). An ideal game should be able to adjust its difficulty dynamically governed by the player's performance. Modern video games utilise a game-testing process to investigate among other factors the perceived difficulty for a multitude of players. In this paper, we investigate how machine learning techniques can be used for automatic difficulty adjustment. Our experiments confirm the potential of machine learning in this application.

1 Introduction

We aim at developing games that provide challenges of the "right" difficulty, i.e., such that players are stimulated but not overburdened. Naturally, what is the right difficulty depends on many factors and can not be fixed once and for all players. For that, we investigate how general machine learning techniques can be employed to automatically adjust the difficulty of games. A general technique for this problem has natural applications in the huge markets of computer and video games but can also be used to improve the learning rates when applied to serious games.

The traditional way in which games are adjusted to different users is by providing them with a way of controlling the difficulty level of the game. To this end, typical levels would be 'beginner', 'medium', and 'hard'. Such a strategy has many problems. On the one hand, if the number of levels is small, it may be easy to choose the right level but it is unlikely that the difficulty is then set in a very satisfying way. On the other hand, if the number of levels is large, it is more likely that a satisfying setting is available but finding it becomes more difficult. Furthermore, choosing the game setting for each of these levels is a difficult and time-consuming task.

In this paper we investigate the use of supervised learning for dynamical difficulty adjustment. Our aim is to devise a difficulty adjustment algorithm that does not bother the actual players. For that, we assume there is a phase of the game development in which the game is played and the difficulty is manually adjusted to be just right. From the data collected in this way, we induce a difficulty model and build it into the game. The actual players do not notice any of this and are always challenged at the difficulty that is estimated to be just right for them.

Our approach to building a difficulty model consists of three steps:

1. cluster the recorded game traces,
2. average the supervision over each cluster, and
3. learn to predict the right cluster from a short period of gameplay.

In order to validate this approach, we use a leave-one-player-out strategy on data collected from a simple game and compare our approach to less sophisticated, yet realistic, baselines. All approaches are chosen such that the players are not bothered. In particular, we want to compare the performance of dynamic difficulty versus constant difficulty as well as the performance of cluster prediction versus no-cluster. Our experimental results confirm that dynamic adjustment and cluster prediction together outperform the alternatives significantly.

2 Motivation and Context

A game and its player are two interacting entities. A typical player plays to have fun, while a typical game wants its players to have fun. What constitutes the *fun* when playing a game?

One theory is that our brains are physiologically driven by a desire to learn something new: new skills, new patterns, new ideas [Biederman and Vessel, 2006]. We have an instinct to play because during our evolution as a species playing generally provided a safe way of learning new things that were potentially beneficial for our life. Daniel Cook [Cook, 2007] created a psychological model of a player as an entity that is driven to learn new skills that are high in perceived value. This drive works because we are rewarded for each new mastered skill or gained knowledge: The moment of mastery provides us with the feeling of joy. The games create additional rewards for their players such as new items available, new areas to explore. At the same

[*] This is a version of "Olana Missura and Thomas Gärtner, *Player Modeling for Intelligent Difficulty Adjustments*. In: Proceedings of the 12th International Conference on Discovery Science (DS) (2009)"

time there are new challenges to overcome, new goals to achieve, and new skills to learn, which creates a loop of learning-mastery-reward and keeps the player involved and engaged.

Thus, an important ingredient of the games that are fun to play is providing the players with the challenges corresponding to their skills. It appears that an inherent property of any challenge (and of the learning required to master it) is its difficulty level. Here the difficulty is a subjective factor that stems from the interaction between the player and the challenge. The perceived difficulty is also not a static property: It changes with the time that the player spends learning a skill.

To complicate things further, not only the perceived difficulty depends on the current state of the player's skills and her learning process, the dependency is actually bidirectional: The ability to learn the skill and the speed of the learning process are also controlled by how difficult the player perceives the task. If the bar is set too high and the task appears too difficult, the player will end up frustrated and will give up on the process in favour of something more rewarding. Then again if the challenge turns out to be too easy (meaning that the player already possesses the skill necessary to deal with it) then there is no learning involved, which makes the game appear boring.

It becomes obvious that the game should provide the challenges for the player of the "right" difficulty level: The one that stimulates the learning without pushing the players too far or not enough. Ideally then, the difficulty of any particular instance of the game should be determined by who is playing it at this moment.

Game development process usually includes multiple testing stages, where a multitude of players is requested to play the game to provide data and feedback. This data is analysed to tweak the games parameters in an attempt to provide a fair challenge for as many players as possible. The question we investigate in this work is how the data from the α/β tests can be used for the intelligent difficulty settings with the help of machine learning.

We proceed as follows: After reviewing related work in Section 3, we describe the algorithm for the dynamic difficulty adjustment in general terms in Section 4. In Sections 5 and 6 we present the experimental setup and the results of the evaluation before concluding in Section 7.

3 Related Work

In the games existing today we can see two general approaches to the question of difficulty adjustment. The traditional way is to provide a player with a way to set up the difficulty level for herself. Unfortunately, this method is rarely satisfactory. For game developers it is not an easy task to map a complex gameworld into a single parameter. When constructed, such a mapping requires additional extensive testing, creating time and money costs. Consider also the fact that generally games require several different skills to play them. The necessity of going back and forth between the gameplay and the settings when the tasks become too difficult or too easy disrupts the flow component of the game.

An alternative way is to implement a mechanism for dynamic difficult adjustment (DDA). One quite popular approach to DDA is a so called *Rubber Band AI*, which basically means that the player and her opponents are virtually held together by a rubber band: If the player is "pulling" in one direction (playing better or worse than her opponents), the rubber band makes sure that her opponents are "pulled" in the same direction (that is they play better or worse respectively). While the idea that the better you play the harder the game should be is sound, the implementation of the Rubber Band AI often suffers from disbalance and exploitability.

There exist a few games with a well designed DDA mechanism, but all of them employ heuristics and as such suffer from the typical disadvantages (being not transferable easily to other games, requiring extensive testing, etc). What we would like to have instead of heuristics is a universal mechanism for DDA: An online algorithm that takes as an input (game-specific) ways to modify difficulty and the current player's in-game history (actions, performance, reactions, ...) and produces as an output an appropriate difficulty modification.

Both artificial intelligence researchers and the game developers community display an interest in the problem of automatic difficulty scaling. Different approaches can be seen in the work of R. Hunicke and V. Chapman [Hunicke and Chapman, 2004], R. Herbich and T. Graepel [Herbrich *et al.*, 2006], Danzi et al [Danzi *et al.*, 2003], and others. As can be seen from these examples the problem of dynamic difficulty adjustment in video games was attacked from different angles, but a unifying approach is still missing.

Let us reiterate that as the perceived difficulty and the preferred difficulty are subjective parameters, the DDA algorithm should be able to choose the "right" difficulty level in a comparatively short time for any particular player. It makes sense, therefore, to conduct the learning in the offline manner and to make use of the data created during the test phases to construct the player models. These models can be used afterwards to generalise to the unseen players.

Player modeling in computer games is a relatively new area of interest for the researchers. Nevertheless, existing work [Yannakakis and Maragoudakis, 2005; Togelius *et al.*, 2006; Charles and Black, 2004] demonstrates the power of utilising the player models to create the games or in-game situations of high interest and satisfaction for the players.

In the following section we present an algorithm that learns a mapping from different player types to the difficulty adjustments and predicts an appropriate one given a new player.

4 Algorithm

To simplify the problem we assume that there exists a finite number of types of players, where by type we mean a certain pattern in behaviour with regard to challenges. That is certainly true, since we have a finite amount of players altogether, possibly times a finite amount of challenges, or timesteps in a game. However, this realistic number is too large to be practical and certainly not fitting the purpose here. Therefore, we discretize the space of all possible players' behaviours to get something more manageable. The simplest such discretization would be into beginners, averagely skilled, and experts (corresponding to easy, average, and difficult settings).

In our experiments we do not predefine the types, but rather infer them using the clustering of the collected data. Instead of attempting to create a universal mechanism for a game to adapt its difficulty to a particular player, we focus on the question of how a game can adapt to a particular player type given two sources of information:

1. the data collected from the alpha/beta-testing stages (offline phase);

2. the data collected from the new player (online phase).

The idea is rather simple. By giving the testers control over the difficulty settings in the offline phase the game can learn a mapping from the set of types into the set of difficulty adjustments. In the online phase, given a new player, the game needs only to determine which type he belongs to and then apply the learned model. Therefore, the algorithm in general consists of the following steps:

1. Given data about the game instances in the form of time sequences
$$T_k = ((t_1, f_1(t_1), \dots, f_L(t_1)), \dots, \\ (t_N, f_1(t_N), \dots, f_L(t_N))),$$
where t_i are the time steps and $f_i(t_j)$ are the values of corresponding features, cluster it in such a way that instances exhibiting similar player types are in the same cluster.
2. Given a new player, decide on which cluster he belongs to and predict the difficulty adjustment using the corresponding model.

Note that it is desirable to adapt to the new player as quickly as possible. To this purpose we propose to split the time trace of each game instance into two parts:

- a prefix, the relatively short beginning that is used for the training of the predictor in the offline phase and the prediction itself in the online phase;
- a suffix, the rest of the trace that is used for the clustering.

In our experiments we used the K-means algorithm [Hartigan and Wong, 1979] for the clustering step and an SVM with a gaussian kernel function [Cortes and Vapnik, 1995] for the prediction step of the algorithm outlined above.

We considered the following approaches to model the adjustment curves in the clusters:

1. The constant model. Given the cluster, this function averages over all instances in the cluster and additionally over the time, resulting in a static difficulty adjustment.
2. The regression model. Given the cluster, we train the regularised least squares regression [Rifkin, 2002] with the gaussian kernel on its instances.

The results stemming from using these models are described in Section 6.

5 Experimental Setup

To test our approach we implemented a rather simple game using the Microsoft XNA framework [1] and one of the tutorials from the XNA Creators Club community, namely "Beginner's Guide to 2D Games" [2]. The player controls a cannon that can shoot cannonballs. The gameplay consists of shooting down the alien spaceships while they are shooting at the cannon (Figure 1). A total of five spaceships can be simultaneously on the screen. They appear on the right side of the game screen and move on a constant height from the right to the left. The spaceships are generated so that they have a random speed within a specific δ-interval from a given average speed. Whenever one of the spaceships is shot down or leaves the game screen, a new one is generated. At the beginning of the game the player's cannon has a certain amount of hitpoints, which is reduced by one every time the cannon is hit. At random timepoints a repair kit appears on the top of the screen, floats down, and disappears again after a few seconds. If the player manages to hit the repair kit, the cannon's hitpoints are increased by one. The game is over if the hitpoints are reduced to zero or a given time limit of 100 seconds is up.

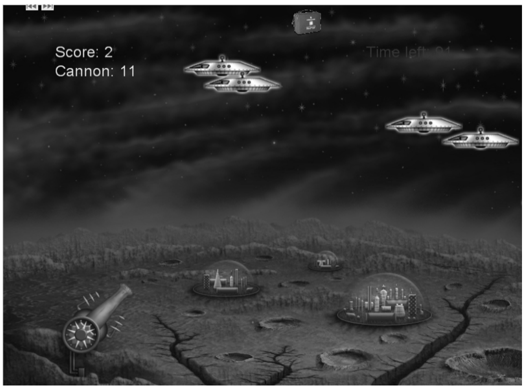


Figure 1: A screenshot showing the gameplay.

Additionally to the controls that allow the player to rotate the cannon and to shoot, there are also two buttons by pressing which the player can increase or decrease the difficulty at any point in the game. In the current implementation the difficulty is controlled by the average speed of the alien ships. For every destroyed spaceship the player receives a certain amount of score points, which increases quadratically with the difficulty level. During each game all the information concerning the game state (e.g. the amount of hitpoints, the positions of the aliens, the buttons pressed, etc) is logged together with a timestamp. At the current state of our work we held one gaming session with 17 participants and collected the data on how the players behave in the game.

Out of all logged features we restricted our attention to the three: the difficulty level, the score, and the health, as they seem to represent the most important aspects of the player's state. The log of each game instance k is in fact a time trace
$$T_k = ((t_1, f_1(t_1), \dots, f_L(t_1)), \dots, \\ (t_N, f_1(t_N), \dots, f_L(t_N))),$$
where $t_1 = 0$, $t_N \leq 100$, and $f_i(t_j)$ is the value of a corresponding feature (Figure 2). Therefore, to model the players we cluster provided by the testers time sequences.

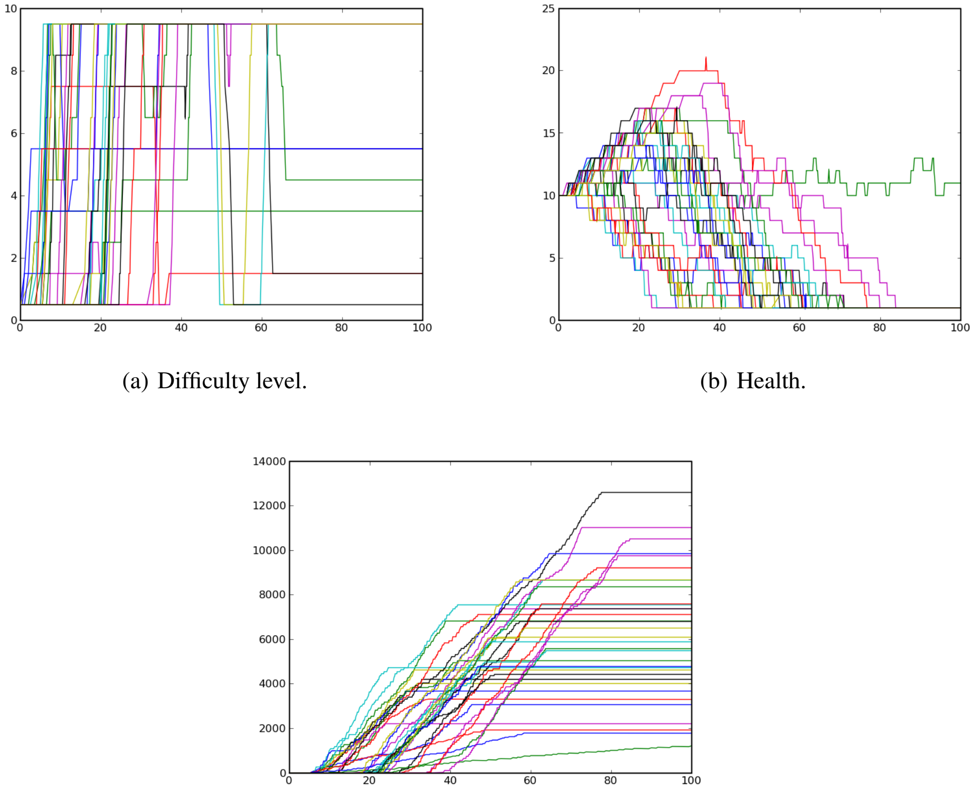
5.1 Technical considerations

Several complications arise from the characteristics of the collected data:

1. Irregularity of the time steps. To reduce the computational load the data is logged only when the game's or the player's state changes (in the case of a simple game used by us it may seem a trivial concern, but this is important to consider for the complex games). As a result for two different game instances k and $\hat{k}$ the time lines will be different:
$$t_{ik} \neq t_{i\hat{k}}.$$

[1] http://msdn.microsoft.com/en-us/xna/default.aspx

[2] http://creators.xna.com/en-GB/



(a) Difficulty level.

(b) Health.

(c) Score.

Figure 2: Game traces from one player. Different colours represent different game instances.

2. Irregularity of the traces' durations. Since there are two criteria for the end of the game (health dropped to zero or the time limit of a hundred seconds is up), the durations of two different game instances k and $\hat{k}$ can be different:

$$t_{Nk} - t_{\hat{N}\hat{k}} \neq 0.$$

The second problem may appear irrelevant, but as described below it needs to be taken care of in order to create a nice, homogeneous set of data points to cluster.

To overcome the irregularity of the time steps we will construct a fit for each trace and then interpolate the data using the fit for every 0.1 of a second to produce the time sequences with identical time steps:

$$\begin{aligned} T_{k fitted} = ((&t_1, f_1(t_1), \dots, f_L(t_1)), \dots, \\ &(t_N, f_1(t_N), \dots, f_L(t_N))), \end{aligned}$$

where $t_1 = 0$, $t_N \leq 100$, and for each $i \in [2, N]$ $t_i = t_{i-1} + 0.1$.

Now it becomes clear why we require the time traces to have equal durations. Since the longest game instances last for a hundred seconds, we need to be able to sample from all of the interpolated traces in the interval between zero and a hundred seconds to create a homogeneous data set. If the original trace was shorter than a hundred seconds, the resulting fitting function wouldn't necessarily provide us with the meaningful data outside of its duration region. Therefore, we augment original game traces in such a way that they all last for a hundred seconds, but the features retain their last achieved values (from the "game over" state):

$$\begin{aligned} T_k = ((&t_1, f_1(t_1), \dots, f_L(t_1)), \dots, \\ &(t_N, f_1(t_N), \dots, f_L(t_N)), \\ &(t_{N+1}, f_1(t_N), \dots, f_L(t_N)), \dots, \\ &(100, f_1(t_N), \dots, f_L(t_N))). \end{aligned}$$

As mentioned in Section 4, after the augmenting step each time trace is split into two parts:

$$\begin{aligned} T_{k pre} &= ((t_1, f_1(t_1), \dots, f_L(t_1)), \dots, \\ &\quad (t_K, f_1(t_K), \dots, f_L(t_K))), \\ T_{k post} &= ((t_{K+1}, f_1(t_{K+1}), \dots, f_L(t_{K+1})), \dots, \\ &\quad (t_N, f_1(t_N), \dots, f_L(t_N))), \end{aligned}$$

where t_K is a predefined constant, in our experiments set to 30 seconds, that determines for how long the game observes the player before making a prediction. The *pre* parts of the traces are used for training and evaluating the predictor. The *post* parts of the traces are used for clustering.

6 Evaluation

To evaluate the performance of the SVM predictor we conduct a kind of "leave one out" cross-validation on the data. For each player presented we construct a following train/test split:

- training set consists of the game instances played by all players except this one;
- test set consists of all the game instances played by this player.

Constructing the train and test sets in this way models a real-life situation of adjusting the game to a previously unseen player. As a performance measure we use the mean absolute difference between the exhibited behaviour in the test instances and the behaviour described by the model of the predicted cluster. The mean is calculated over the test instances.

To provide the baselines for the performance evaluation, we construct for each test instance a sequence of "cheating" predictors: The first (best) one chooses a cluster that delivers a minimum possible absolute error (that is the difference between the predicted adjustment curve and the actual difficulty curve exhibited by this instance); the second best chooses the the cluster with the minimum possible absolute error from the remaining clusters, and so on. We call these predictors "cheating" because they have access to the test instances' data before they make the prediction. For each "cheating" predictor the error is averaged over all test instances and the error of the SVM predictor is compared to these values. As the result we can make some conclusion on which place in the ranking of the "cheating" predictors the SVM one takes.

Figure 3 illustrates the performance of the SVM predictor and the best and the worst baselines for a single player and 7 clusters. We can see from the plots that for each model the SVM predictor displays the performance close to the best cluster. Figure 4 shows that the performance of the SVM predictor averaged over all train/test splits demonstrates similar behaviour.

Statistical Tests

To verify our hypotheses, we performed proper statistical tests with the null hypothesis that the algorithms perform equally well. As suggested recently [Demšar, 2006] we used the Wilcoxon signed ranks test.

The Wilcoxon signed ranks test is a nonparametric test to detect shifts in populations given a number of paired samples. The underlying idea is that under the null hypothesis the distribution of differences between the two populations is symmetric about 0. It proceeds as follows:

1. compute the differences between the pairs,
2. determine the ranking of the absolute differences, and
3. sum over all ranks with positive and negative difference to obtain W_+ and W_-, respectively.

The null hypothesis can be rejected if W_+ (or $\min(W_+, W_-)$, respectively) is located in the tail of the null distribution which has sufficiently small probability.

For settings with a reasonably large number of measurements, the distribution of W_+ and W_- can be approximated sufficiently well by a normal distribution. Unless stated otherwise, we consider the 5% significance level ($t_0 = 1.78$).

Dynamic versus Static Difficulty

We first want to confirm the hypothesis that a dynamic difficulty function is more appropriate than a static one. To eliminate all other influences, we considered first and foremost only a single cluster. In this case, as expected, the dynamic adjustment significantly outperforms the static setting ($t = 2.67$).

We also wanted to compare the performance of dynamic and static difficulty adjustment for larger numbers of clusters. To again eliminate all other influences, we considered the best and the worst "cheating" predictor for either strategy. The t-values for these comparisons are displayed in Table 1.

Table 1: t-values for comparison of the constant model vs the regression model for the varying amount of clusters.

c	best-const vs best-regr	worst-const vs worst-regr
1	8.46	8.46
2	6.12	9.77
3	5.39	12.64
4	5.26	11.37
5	4.90	12.62
6	4.77	11.05
7	4.80	10.38
8	4.62	6.83
9	4.61	7.20
10	4.63	4.36
11	4.55	0.71
12	4.68	-0.77
13	4.60	-9.16
14	4.50	-5.54
15	4.57	-13.26

While varying the amount of clusters from one to fifteen we found out that dynamic difficulty adjustment (the regression model) always significantly outperforms the static one (the constant model) for choosing the best cluster. The same effect we can observe for the worst predictor, but only until the amount of clusters used is greater than ten. For more clusters the static model starts to outperform the dynamic one, probably due to there being insufficient amount of instances in some clusters to train a good regression model. Based on these results in the following we consider only the regression model and vary the amount of clusters from one to ten.

Right versus Wrong Choice of Cluster

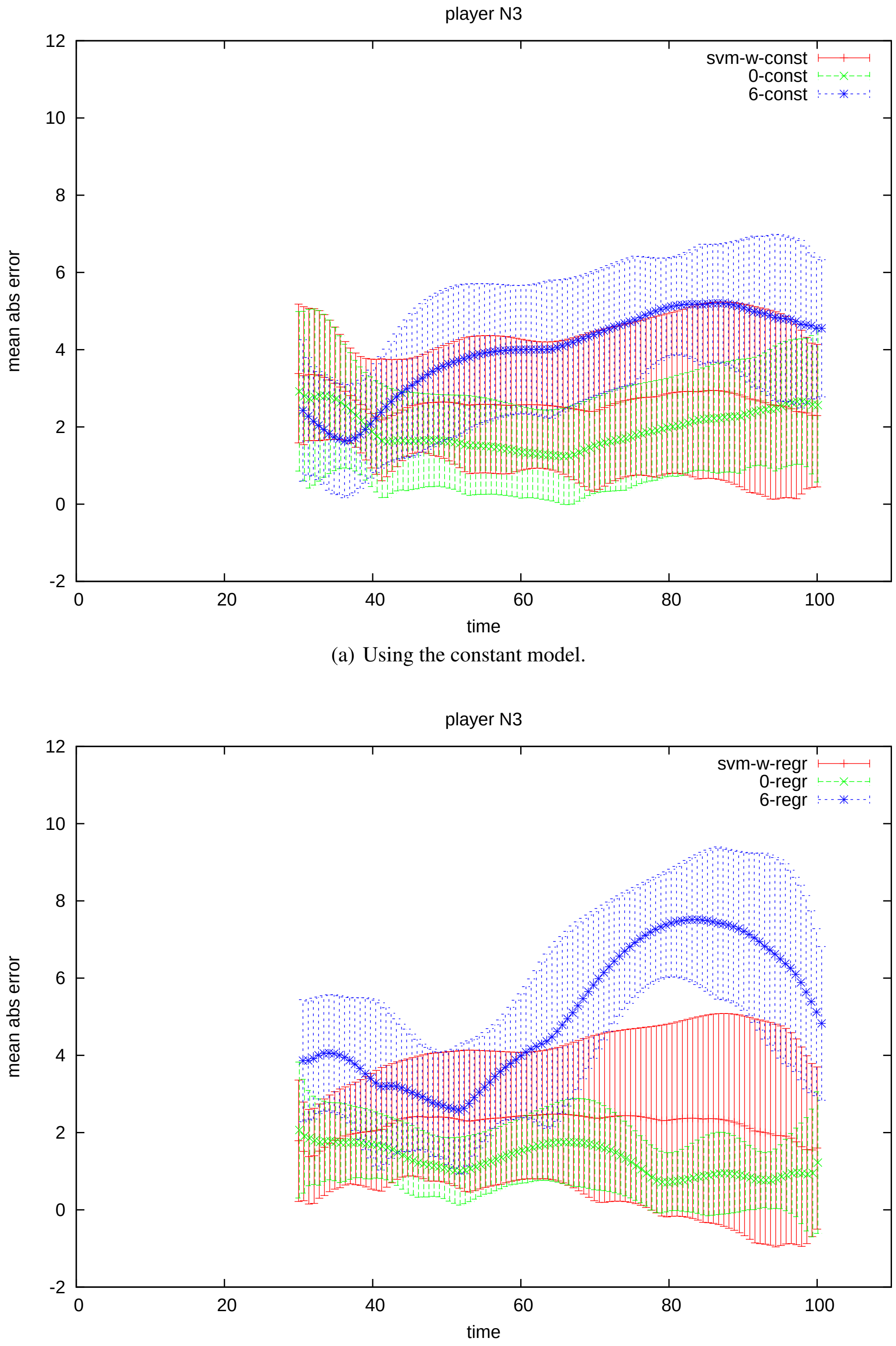
As a sanity check, we next compared the performance of the best choice of a cluster versus the worst choice of cluster. To this end we found—very much unsurprisingly—that for any non-trivial number of clusters, the best always significantly outperforms the worst.

This means there is indeed room for a learning algorithm to fill. The best we can hope for is that in some settings the performance of the predicted cluster is close to, i.e., not significantly worse than, the best predictor while always being much, i.e., significantly, better than the worst predictor.

One versus Many Types of Players

The last parameter that we need to check before coming to the main part of the evaluation is the number of clusters. It can easily be understood that the quality of the best static model improves with the number of clusters while the quality of the worst degrades even further. Indeed, on our data, having more clusters was always significantly better than having just a single cluster for the best predictor using the regression model.

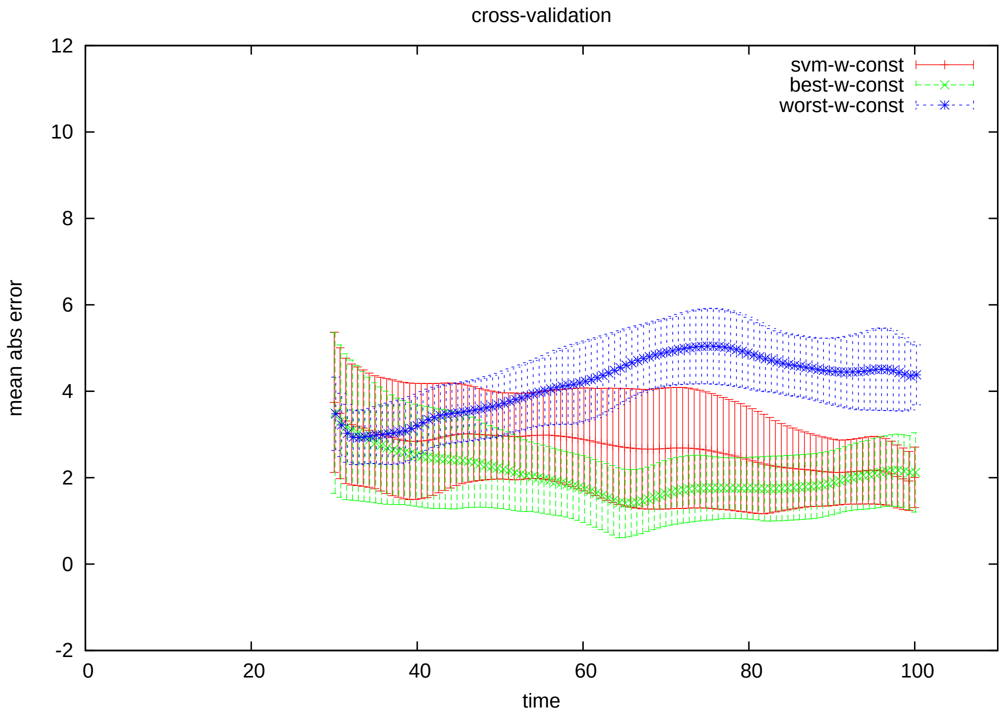
Under the assumption that we do not want to burden the players with choosing their difficulty, this implies that we do need a clever way to automatically choose the type of the player. Adjusting the game just to a single type is not sufficient.



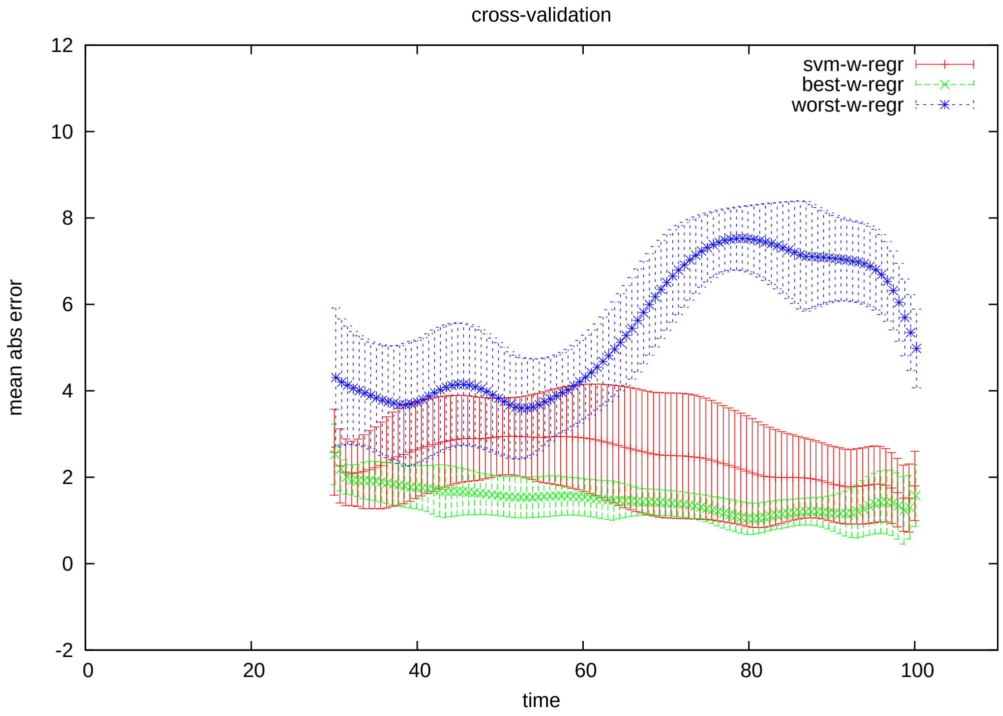
(a) Using the constant model.

(b) Using the regression model.

Figure 3: An example of the predictors' performances for one player.



(a) Using the constant model.



(b) Using the regression model.

Figure 4: The predictors' performance averaged over all train/test splits for 7 clusters.

Table 2: Results of the significance tests for the comparison of performance of the SVM predictor and "cheating" predictors using the regression model.

	1	2	3	4	5	6	7	8	9	10
1	s									
2	w	b								
3	w	s	b							
4	w	b	b	b						
5	w	s	b	b	b					
6	w	w	b	b	b	b				
7	w	s	b	b	b	b	b			
8	w	s	b	b	b	b	b	b		
9	w	w	s	b	b	b	b	b	b	
10	w	w	s	b	b	b	b	b	b	b

Quality of Predicted Clusters

We are now ready to consider the main evaluation of how well the type of the player can be chosen automatically. As mentioned above the best we can hope for is that in some settings the performance of the predicted cluster is close to the best cluster while always being much better than the worst cluster. Another outcome that could be expected is that performance of the predicted cluster is far from that of the best cluster as well as from the worst cluster.

To illustrate the quality of the SVM predictor we look at its place in the ranking of the "cheating" predictors while varying the amount of clusters. The results of the comparison of the predictors' performance for the regression model are shown in Table 2. Each line in the table corresponds to the amount of clusters specified in the first column. The following columns contain values 'w', 's', and 'b', where 'w' means that the SVM predictor displayed the significantly worse performance than the corresponding "cheating" predictor, 'b' for the significantly better performance, and 's' for the the cases where there was no significant difference. The columns are ordered according to the ranking of the "cheating" predictors, i.e. 1 stands for the best possible predictor, 2 for the second best, and so on.

We can observe a steady trend in the SVM predictor's performance: Even though it is always (apart from the trivial case of one cluster) significantly worse than that of the best possible predictor, it is also always significantly better than that of the most other predictors. In other words, regardless of the amount of clusters, the SVM predictor always chooses a reasonably good one.

This last investigation confirms our hypothesis that predicting the difficulty-type for each player based on short periods of gameplay is a viable approach to taking the burden of choosing the difficulty from the players.

7 Conclusion and Future Work

In this paper we investigated the use of supervised learning for dynamical difficulty adjustment. Our aim was to devise a difficulty adjustment algorithm that does not bother the actual players. Our approach to building a difficulty model consists of clustering different types of players, finding a good difficulty adjustment for each cluster, and predicting the cluster for short traces of gameplay. Our experimental results confirm that dynamic adjustment and cluster prediction together outperform the alternatives significantly.

One parameter left out in our investigation is the length of the prefix that is used for the prediction. We will investigate its influence on the predictors' performance in the future work. We also plan to collect and examine more players' data to see how transferable our algorithm is to the other games. Another direction for the future investigation is the comparison of our prediction model to the other algorithms employed for the time series predictions, such as neural networks or gaussian processes.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful and insightful comments.

References

[Biederman and Vessel, 2006] Irving Biederman and Edward Vessel. Perceptual pleasure and the brain. *American Scientist*, 94(3), 2006.

[Charles and Black, 2004] D. Charles and M. Black. Dynamic player modeling: A framework for player-centered digital games. In *Proc. of the International Conference on Computer Games: Artificial Intelligence, Design and Education*, pages 29–35, 2004.

[Cook, 2007] Daniel Cook. The chemistry of game design. Gamasutra, 07 2007.

[Cortes and Vapnik, 1995] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[Danzi *et al.*, 2003] G. Danzi, A. H. P. Santana, A. W. B. Furtado, A. R. Gouveia, A. Leitão, and G. L. Ramalho. Online adaptation of computer games agents: A reinforcement learning approach. *II Workshop de Jogos e Entretenimento Digital*, pages 105–112, 2003.

[Demšar, 2006] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1), 2006.

[Hartigan and Wong, 1979] JA Hartigan and MA Wong. A k-means clustering algorithm. *JR Stat. Soc., Ser. C*, 28:100–108, 1979.

[Herbrich *et al.*, 2006] Ralf Herbrich, Tom Minka, and Thore Graepel. Trueskill$^{\text{tm}}$: A bayesian skill rating system. In *NIPS*, pages 569–576, 2006.

[Hunicke and Chapman, 2004] R. Hunicke and V. Chapman. AI for dynamic difficulty adjustment in games. *Proceedings of the Challenges in Game AI Workshop, Nineteenth National Conference on Artificial Intelligence*, 2004.

[Rifkin, 2002] R. M. Rifkin. *Everything Old is new again: A fresh Look at Historical Approaches to Machine Learning*. PhD thesis, MIT, 2002.

[Togelius *et al.*, 2006] J. Togelius, RD Nardi, and S.M. Lucas. Making racing fun through player modeling and track evolution. In *SAB06 Workshop on Adaptive Approaches for Optimizing Player Satisfaction in Computer and Physical Games*, pages 61–70, 2006.

[Yannakakis and Maragoudakis, 2005] G.N. Yannakakis and M. Maragoudakis. Player Modeling Impact on Player's Entertainment in Computer Games. *Lecture notes in computer science*, 3538:74, 2005.

Named Entity Resolution Using Automatically Extracted Semantic Information

Anja Pilz and **Gerhard Paaß**
Fraunhofer Institute Intelligent Analysis and Information Systems
St. Augustin, Germany
{anja.pilz, gerhard.paass}@iais.fraunhofer.de

Abstract

One major problem in text mining and semantic retrieval is that detected entity mentions have to be assigned to the true underlying entity. The ambiguity of a name results from both the polysemy and synonymy problem, as the name of a unique entity may be written in variant ways and different unique entities may have the same name. The term "bush" for instance may refer to a woody plant, a mechanical fixing, a nocturnal primate, 52 persons and 8 places covered in Wikipedia and thousands of other persons. For the first time, according to our knowledge we apply a kernel entity resolution approach to the German Wikipedia as reference for named entities. We describe the context of named entities in Wikipedia and the context of a detected name phrase in a new document by a context vector of relevant features. These are designed from automatically extracted topic indicators generated by an LDA topic model. We use kernel classifiers, e.g. rank classifiers, to determine the right matching entity but also to detect uncovered entities. In comparison to a baseline approach using only text similarity the addition of topics approach gives a much higher f-value, which is comparable to the results published for English. It turns out that the procedure also is able to detect with high reliability if a person is not covered by the Wikipedia.

1 Introduction

The problem of name ambiguity exists in many forms. It is common for different people to share the same name. For example, there is a Gerhard Schröder who was chancellor of the Federal Republic of Germany, another who was Federal Minister in Germany, and several more who are broadcaster officials or journalists. Locations may have the same name. For example, there are 27 municipalities in Germany called Neustadt. The acronyms associated with organizations may also be ambiguous. UMD may stand for the University of Michigan – Dearborn, the University of Minnesota, Duluth or the University of Maryland.

On the other hand there may be different names for the same entity. For many organizations there exist a number of different acronyms and designations. For the political party Sozialdemokratische Partei Deutschlands we have the synonyms Sozialdemokraten, SPD, Sozis, etc.

The effects of name ambiguity can be seen when carrying out web searches or retrieving articles from an archive of newspaper text. For example, the top 10 hits of a Google search for "Peter Müller" mention seven different people. While it may be clear to a human that the Prime Minister from Saarland, the boxer from Cologne, and the professor of mathematics of the university of Würzburg are not the same person, it is difficult for a computer program to make the same distinction. In fact, a human may have a hard time retrieving all the material relevant to the particular person they are interested in without being swamped by information on namesakes.

Approaches to entity resolution generally rely on the strong contextual hypothesis of Miller and Charles [Miller and Charles, 1991], who hypothesize that words with similar meanings are often used in similar contexts. This is equally true for proper names, where a particular entity will likely be mentioned in certain contexts. For example, Peter Müller the prime minister may not be mentioned with Würzburg University very often, while Peter Müller the Professor will be. Thus, our approach to entity resolution consists in finding classes of similar contexts such that each class represents a distinct entity.

As a first step we identify *name phrases*, which may refer to *named entities* like persons, organizations, locations etc. There are a number of quite mature techniques for *name phrase recognition*, such as persons or locations [Sang and Meulder, 2003]. Customary they are termed named entity recognizers, although they do only spot possible name phrases.

A second step is the *entity resolution*, the identification of the identity of each name phrase. More formally we want to assign a name phrase n and the associated context information $c(n)$, e.g. the surrounding words, to one of the candidate entities in a set $\mathcal{E}_n = \{e_1, \ldots, e_m\}$ where each e_i corresponds to a "true" underlying named entity, e.g. a specific person. Each entity e_i in turn is characterized by features $d(e_i)$, e.g. the words in a description of e_i. In this paper we are mainly interested in assigning name phrases to the corresponding Wikipedia article, which can be considered as an unambiguous reference for a specific entity.

Note that we have to find out, whether a person is covered in Wikipedia at all. This is a difficult task, as, for example, there are 583 persons with name Gerhard Schröder listed in the German telephone directory, whereas only 5 are mentioned in Wikipedia. If it is not possible to assign n to one of the know entities in $\mathcal{E}$ we may formally assign it to e_{out}.

In the next section we describe related work to entity resolution. Then we outline the properties of Wikipedia as a reference for unique named entities. Subsequently we describe the kernel approach used in this paper. Then we describe the experimental setup for training and testing. Then we describe the results and summarize the paper.

2 Related Work

Named entity resolution is closely related to the task of word sense disambiguation (WSD) aiming to resolve the ambiguity of common words in a text. Both tasks have in common, that a meaning of a mention is strongly dependent on the context it appears in. Entity resolution, however, usually concerns only phrases of a restricted type (e.g. persons) but there are potentially very many underlying target entities. Nevertheless, many concepts of WSD may be applied to entity resolution.

2.1 Unsupervised Approaches

One of the first works in the field of entity resolution was that of [Bagga and Baldwin, 1998]. In their approach they create context vectors for each occurrence of a name. Each vector contains exactly the words that occur within a fixed-sized window around the ambiguous name n. The similarity among names is measured using the cosine measure. To evaluate their approach they created the ”John Smith” corpus, which consists of 197 new articles mentioning 35 different John Smiths.

Agglomerative clustering is used by [Gooi and Allan, 2004] to form groups of context vectors. Among others they employed the Kullback-Leibler distance measure. They conclude that agglomerative clustering works particularly well, and observe that a context window size of 55 words gives optimum performance. [Mann and Yarowsky, 2003] enhance the representation of context by biographic facts (date/place of birth, occupation, etc.) extracted from the web using pattern matching. They show that these automatically extracted biographic information to improves clustering results, which are the basis for entity resolution.

[Bekkerman and McCallum, 2005] present two unsupervised approaches to disambiguate persons mentioned in Web pages. One method exploits that Web pages of people related to each other are likely to be interconnected. The other approach simultaneously clusters documents and words employing the fact that similar documents have similar distributions over words, while similar words are similarly distributed over documents. The derived word similarity as well as the link features are successfully used for named entity resolution.

[Bhattacharya and Getoor, 2005] extensively use relational evidence for entity resolution. In the context of citations we may conclude that ”R. Srikant” and ”Ramakrishnan Srikant” are the same author, since both are coauthors of another author. They consider the mutual relations between authors, paper titles, paper categories, and conference venues. They argue that if they jointly resolve the identity of authors, papers, etc. that this leads to a better result than considering each type alone. They construct probabilistic networks capitalizing on two main ideas: First they use tied parameters for repeating features over pairs of references and their resolution decisions. Second they exploit the overlap between decisions, as two different decisions are dependent. They use similarity measures to resolve entities by clustering them taking into account the relations between objects, and get encouraging results.

2.2 Supervised Approaches

A classifier is used by [Han *et al.*, 2004] to resolve entities in citations. Their naive Bayes approach takes authors as classes, computes the prior probability of each author, and uses coauthors, title words, and publishing details extracted from the citation as features. Their SVM method again considers each author as a class using the same features.

[Hassel *et al.*, 2006] disambiguate researcher names in citations by exploiting relational information contained in an ontology derived from the DBLP database. Attributes such as affiliations, topics of interests, or collaborators are extracted from the ontology and matched against the text surrounding a name occurrence. The results of the match are then combined in a linear scoring function that ranks all possible senses of that name. This scoring function is trained on a set of disambiguated name queries that are automatically extracted from Wikipedia articles. The method is also able to detect when a name denotes an entity that is not covered in Wikipedia.

[Huang *et al.*, 2006] combine supervised distance measure learning and unsupervised clustering and apply it to author disambiguation. The distance metric between papers used in DBSCAN is calculated by an online active selection support vector machine algorithm (LASVM), yielding a simpler model, lower test errors and faster prediction time than a standard SVM.

[Bunescu and Pasca, 2006] resolve entities using a specific version of the SVM which generates a ranked list of plausible entities. This ranking SVM was introduced by [Joachims, 2002]. As features they use all words in a window around the name phrase as well as the Wikipedia categories. For training Wikipedia articles are used as unambiguous references for specific entities. They are described by the words and categories of these articles. Within Wikipedia most articles are linked to specific spots in other articles. The text around the name phrase in a link is used as an instance for the occurrence. In this paper we adapt this approach to entity resolution for German name phrases. [Wentland *et al.*, 2008] expands this method to other languges.

[Cucerzan, 2007] present a large-scale system for the recognition and semantic disambiguation of named entities based on information extracted from Wikipedia and Web search results. The system uses coreference analysis to associate different surface forms of a name in a text, e.g. ”George W. Bush” and ”Bush”. In addition to context words they use Wikipedia categories to describe an entity. Within Wikipedia they use the article about an entity as well as the context of links to an entity to characterize an entity. By the links in Wikipedia they get multiple contexts of an entity in Wikipedia and by coreference resolution they get multiple contexts for a name phrase in a new document. The assignment is done by maximizing the non-normalized scalar products for the contexts of entities and name phrases.

[Waltinger and Mehler, 2008] exploit the context-surrounding of a token by analyzing the border-sentences of an entity. The information of the incorporated context is used for building an expanded context graph around the unknown entity. This is done by querying a co-occurrence network, keeping the most significant edges to the context-instance. The approach shows very promising results.

3 Wikipedia as Reference Knowledge Resource

Wikipedia is a web-based, free content encyclopedia project, written collaboratively by volunteers using a wiki software that allows almost anyone to add and change articles. Since its creation in 2001 it has become the largest organized knowledge repository on the Web. In this paper we use the German version. In Nov. 2008 it had 845k articles with an average length of 3500 bytes and 20.8 million internal links [Wikimedia, 2009].

Articles hold information focused on one specific entity or concept. An article is uniquely identified by the common name for the subject or person described in the article. Ambiguous names are further qualified with additional information placed in parentheses. For instance, the six entities sharing the name Michael Müller are distinguished by their affiliations, occupations, or locations: Michael Müller (Berlin), Michael Müller (Comedian), Michael Müller (FDP), Michael Müller (Handballspieler). Michael Müller (Liedermacher), and Michael Müller (SPD).

Every article in Wikipedia has one or more categories, representing the topics it belongs to. Categories can be very broad but also very specific, i.e. applying only to two persons such as the category ”Träger des Bundesverdienstkreuzes (Großkreuz in besonderer Ausführung)”. Relations among entity and concepts are expressed by links. When mentioning an entity or concept with an existing article page, contributing authors are required to link at least the first mention to the corresponding article. This link structure may be used to estimate semantic relatedness [Milne, 2007].

4 Semantic Information from Topic Models

Wikipedia contains several million different words and usual similarity metrics such as the cosine similarity are not capable to grasp the synonymy of different terms. The same content describing entities may be expressed by completely different words, i.e. ”Bundeskanzler” and ”Regierungschef” , which shows that the direct comparison of words, even in a stemmed form, may be misleading.

4.1 Topic Modeling by Latent Dirichlet Allocation

Topic modeling by Latent Dirichlet Allocation [Blei *et al.*, 2003] aims to represent the meaning of sentences and documents by a low-dimensional vector of ”topics”. It assumes the following simplified probabilistic model for the generation of a document d_i:

- Randomly generate the number of words in a document $N \approx Poisson(\xi)$, where ξ is a fixed prior parameter.
- Randomly choose a h-dimensional probability vector describing the distribution of topics in the document $\theta \sim Dirichlet(\alpha)$.
- For each of the N words w in the document
  - Randomly choose a topic $z_k \sim Multinomial(\theta)$, where α is a h-dimensional parameter vector describing the prior distribution of probability values.
  - Randomly select a word $w_n \sim p(w_n|z_k, \beta)$, where the multinomial distribution $p(w_n|z_k, \beta)$ describes the probability of words for the topic z_k.

Given a training set of unlabeled documents all free parameters in the model, the conditional distribution $p(w_n|z_n, \beta)$ of word given topics as well as the hyperparameters ξ,α and β may be estimated. Using a Bayesian approach to regularize parameters [Blei *et al.*, 2003] propose an efficient approximate inference techniques based on variational methods. The resulting word distributions $p(w_n|z_n, \beta)$ for each topic have high probabilities for words that often co-occur in documents. Topics alleviate two main problems arising in natural languages: synonymy and polysemy. Synonymy refers to a case where two different words (say car and automobile) have the same meaning. These synonyms usually will occur in the same topics. Polysemy on the other hand refers to the case where a term such as plant has multiple meanings (industrial plant, biological plant). Depending on the context (industry or biology) different topics will be assigned to the word plant.

A document is generated by picking a distribution over topics (i.e. mostly about DOG, mostly about CAT, or a bit of both), and given this distribution, picking the topic of each specific word. Then words are generated given their topics. (Notice that words are considered to be independent given the topics. This is a standard bag of words model assumption, and makes the individual words exchangeable.)

The application of a topic model to a Wikipedia article has a similar effect as the manual assignment of categories by Wikipedia users. The articles content is given some labels as to whether which topics/categories are the most probable. Generally, one cannot assume that each assignment is relevant since many categories exist, that only apply to two persons and are hence to specific. The assumption that category assignments are appropriate (correct) need not be true but in the end, this is a problem from which topic models suffer as well.

Topic Models can help to solve this problem. They rely on the article text and not on the user's intuition and therefore they are more conservative in the assignment of meanings.

The usage of topic models, e.g. for example the assignment of the highest probability topics as feature to each query, has two reasons. Alternative meanings of words are grasped by the model and hence the similarity measure is less restrictive. Additionally, we gain a summary of the contexts subject.

4.2 Training

The topic model is trained on 100000 Wikipedia articles that have persons as subject. The number of topics is set to 200 which was considered as appropriate given the number of training articles. A manual analysis of models with higher or lower granularity in topics revealed more volatile or less expressive topic clusters.

4.3 Inference

For a trained topic model, one can compute the probability of each of the 200 topics to be present in a new document (see for example [Blei *et al.*, 2003]). We can thus define a new feature for both the query context and the article text. Let

- $e.\mathcal{T}$ be the probability distribution of topics in the article text $e.T$
- $c.\mathcal{T}$ be the probability distribution of topics in the query text $c.T$

assigned by a pre-trained model.

To demonstrate the distinctive character of these attributes, we give the example of a context mentioning an

entity called *Willi Weyer*, with two candidates in Wikipedia, i.e. the politician *Willi Weyer* and the soccer player *Willi Weyer (Fußballspieler)*. The context is extracted from an article on delegates in a German federal state and consists of the following words (stopwords removed):
{*Weyer, Willi, Landeslist, SPD, CDU, Wahlkreis, Heinrich, FDP, Wenk, Detmold, Wendt, Hermann, Geld, Wehr, Wilhelm, Minden-Nord, Wehking, Juni, Landeslist, Wiesmann, Recklinghausen-Land-Sudw, Wint, Friedrich, Lemgo-W, Witthaus, Bernhard, Mulheim-Ruhr-Sud, Wolf*}
All terms are stemmed using the Snowball algorithm for German [Porter, 2001].

The application of the topic model on this text yields a probability distribution that presents the probability for each topic known to the model to be presented in the context and hence new semantic features for the context descritpion, i.e. $c.\mathcal{T} = \{t_{67}, t_{106}, t_9\}$. Table 1 shows the most important words of these topic clusters with the topics probability given the context at hand. In this case, the three most probable topics do not fit the query equally good, as can already be seen from each topics probability value. Whereas the topic with the highest probability indicates an political context, e.g. the entity's occupation, the less probable topics are due to the relatively high frequency of names in the context, that are associated with different name clusters (i.e. names in historical context such as royals).

Topic	The 15 most important words	$P(t_i\|c.T)$
t_{67}	Vorsitz Abgeordnet stellvertret SPD Landtag CDu Bundestag Wahlkreis Ausschuss Oberburgermeist FDP Politikerin Stadtrat Fraktion eingezog	0.255
t_{106}	Karl Heinrich preussisch Ferdinand Wurzburg Landwirtschaft Freiherr Gut Geheim Dom Greifswald landwirtschaft Kuhn Pomm konig	0.0611
t_9	August Friedrich Wilhelm Gross Christian Philipp Elisabeth Adolf geb Katharina Luis Moritz Sophi Rhein Conrad	0.0416

Table 1: Most probable topics for the query of *Willi Weyer*

Candidate entity Willi Weyer: We now compare the entity mentioned in the query to the possible candidates extracted from Wikipedia. The application of the topic model on the entity's article text $e_1.T$ yields again a probability distribution denoting the probability for each topic to be present in the article, which is here summarized into $e_1.\mathcal{T} = \{t_{67}, t_{186}, t_{105}\}$, as also shown in table 2. Obviously, the first two topics represent the entity's occupation very well and the third holds a relation to the fact that *Weyer* established traffic reports and highway police in his federal state. Comparing this information to the Wikipedia categories assigned to the entity (see table 3), it becomes obvious that they relate very well to each other and the automatically extracted feature holds an equal amount of information.

Candidate entity Willi Weyer (Fußballspieler): The other candidate entity is a former German soccer player with the qualified Wikipedia article name *Willi Weyer (Fußballspieler)*. For the associated article text, the three most probable topics yield $e_2.\mathcal{T} = \{t_{168}, t_{122}, t_{148}\}$, as shown in table 4. All three of the assigned topics relate to the entity's occupation and specify it if further by incorporating the teams the entity was engaged with. Considering that this candidate is assigned only one category, i.e. that of *Fußballspieler (Deutschland)*, we can deduce much more information from the assigned topics. Note that the relatively high probabilities deduce a very distinctive association to topics and hence indicate differing contexts that allow us to distinguish among entities merely by the context they appear in.

Topic	The 15 most important words	$P(t_i\|e.T)$
t_{67}	Vorsitz Abgeordnet stellvertret SPD Landtag CDu Bundestag Wahlkreis Ausschuss Oberburgermeist FDP Politikerin Stadtrat Fraktion eingezog	0.124
t_{186}	Prasident Regier Bitt Amtszeit Minist Ministerprasident loesch Erklaerung Kabinett Rucktritt Premierminist Reform Aussenminist Liberal Finanzminist	0.0956
t_{105}	fuhr Renn gefahr Fahr todlich unfall Motor Wag Auto Racing Kreuzzug byzantin Roberto fahr ergriff	0.0713

Table 2: Most probable topics for the article of the politician *Willi Weyer*

Wikipedia categories for Willi Weyer
Finanzminister (Nordrhein-Westfalen)
Landtagsabgeordneter (Nordrhein-Westfalen)
Bundestagsabgeordneter
FDP-Mitglied
Sportfunktionär

Table 3: Wikipedia categories for the politician *Willi Weyer*.

5 Learning Ranking Functions

As in [Joachims, 2002] we start with a collection $D = \{d_1, \ldots, d_m\}$ of articles specifying unique entities. For a context $c = c(n)$ containing features describing a name phrase n, we want to determine a list of relevant articles in D, where the most relevant articles appear first. This corresponds to a ranking relation $r^*(c) \subseteq D \times D$ that fulfills the properties of a weak ordering, i.e. asymmetric and transitive. If a document d_i is ranked higher than d_j for an ordering r, i.e. $d_i <_r d_j$, then $(d_i, d_j) \in r$, otherwise $(d_i, d_j) \notin r$.

We have to measure the similarity of a proposed ranking $r(c)$ and the target ranking $r^*(c)$. Such a measure is Kendall's τ [Kendall, 1955] which is a function of the number n_d of concordant pairs in relation to all pairs. A pair $d_i \neq d_j$ is *concordant* if either $(d_i, d_j) \in r_a \wedge (d_i, d_j) \in r_b$ or $(d_j, d_i) \in r_a \wedge (d_j, d_i) \in r_b$.

Now assume we have training set S containing n different i.i.d. contexts $c_i = c(n_i)$ with their target rankings $(c_1, r_1^*), (c_2, r_2^*), \ldots, (c_n, r_n^*)$, where c_i contains a description of a context and $r_i^* \subseteq D \times D$ is a ranking on the documents at hand. Now the learning algorithm should estimate a ranking $\hat{r}(c) \in D \times D$ for a context $c(n)$ that maximizes the fraction of concordant pairs. A ranking $\hat{r}(c)$ can be defined with a linear ranking function

$$(d_i, d_j) \in r(c(n)) \Longleftrightarrow w'\Phi(c(n), d_i) > w'\Phi(c(n), d_j) \tag{1}$$

where $\Phi(c(n), d_i)$ is a given mapping of the context features $c(n)$ and the features of document d_i into a high-dimensional feature space and w is a weight vector of

Topic	The 15 most important words	$P(t_i\|e.T)$
t_{168}	Saison Tor Fussballspiel Einsatz Mannschaft Bundesliga Sturm schoss Mittelfeldspiel Borussia Aufstieg nationalmannschaft Regionalliga nationaljahr Eintracht	0.1705
t_{122}	Koln Dusseldorf Kurt Rot Aach Nationalsozialist Bernd Willi freien KPD Wuppertal Hubert Rheinland Machtergreif Nordrhein-Westfal	0.0932
t_{148}	Spiel Train erzielt Nationalmannschaft bestritt Landerspiel Fussball Fussballspiel Fussballnationalmannschaft Klub Pokalsieg Treff Fussball-Weltmeisterschaft Nationalspiel Europapokal	0.0685

Table 4: Most probable topics for the article of the soccer player *Willi Weyer (Fußballspieler)*

matching dimension. For the linear ranking functions defined above maximizing the number of concordant pairs is equivalent to finding the weight vector w so that the maximum number of the following inequalities holds:

$$\begin{aligned} \forall_{(d_i,d_j)\in r_1^*} w\Phi(c(n_1),d_i) &> w\Phi(c(n_1),d_j) \quad (2)\\ &\dots\\ \forall_{(d_i,d_j)\in r_n^*} w\Phi(c(n_n),d_i) &> w\Phi(c(n_n),d_j) \end{aligned}$$

As the exact solution of this problem is NP-hard an approximate solution is proposed by [Joachims, 2002] by introducing non-negative slack variables $\xi_{i,j,k}$ and minimizing the sum of slack variables. Regularizing the length of w to maximize margins leads to the following optimization problem

$$\text{minimize } V(w,\xi) = \frac{1}{2}w * w + C\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{n}\xi_{i,j,k} \quad (3)$$

subject to

$$\begin{aligned} &\forall k\forall(d_i,d_j)\in r_k^*:\\ &\quad w\Phi(c(n_k),d_i) \geq w\Phi(c(n_k),d_j)+1-\xi_{i,j,k}(4)\\ &\forall k\forall i\forall j: \quad \xi_{i,j,k} \geq 0 \end{aligned}$$

C is a parameter trading-off the training error in terms of n_d to the margin size. The optimization is convex and has no local optima. As the inequalities are equivalent to $w\left(\Phi(c(n_k),d_i)-\Phi(c(n_k),d_j)\right) \geq 1-\xi_{i,j,k}$ we get the same optimization problem as the usual SVM for difference vectors $\Phi(c(n_k),d_i)-\Phi(c(n_k),d_j)$. The algorithm is implemented in the SVM^{light} package of Thorsten Joachims and similar to that of structured SVMs. As usual non-linear feature mappings for arbitrary kernels may be used.

Note that according to the properties of the SVM, the algorithm should be able to generalize. If the set of training contexts $c_1,\dots,c_n$ is i.i.d. and representative, then the optimal parameter should also give near-optimal rankings for new contexts, which follow the underlying context distribution.

6 Entity Resolution Approaches

In this paper we adapt the entity resolution approach of [Bunescu and Pasca, 2006] and enhance it with additional features. For the first time, according to our knowledge, we apply it to the resolution of German name phrases using German Wikipedia entries. We represent a name phrase n by its context $c(n)$ and want to assign it to one of a set of Wikipedia articles $D=\{d_1,\dots,d_m\}$.

For a name phrase n we determine the set of candidate articles by using the disambiguation pages of Wikipedia. If only the surname is available, this set of candidates is large, while the set is much smaller if n contains a first name and a surname. Currently we ignore errors like misspellings. In principle they may also be taken into account, e.g. by a specific Levenshtein distance measure.

6.1 Ranking with Context-Article Similarity

The first approach to model similarity between a query context and an Wikipedia article context is a simple summation over common words, based on the idea, that the larger this number the more similar the context and hence the more similar the entities denoted. [Bunescu and Pasca, 2006] and [Cucerzan, 2007] both evaluated experimentally a ranking function based on the cosine similarity between the context of the query and the text of the entity's article:

$$\phi_{cos} = \cos(c.T, e.T) = \frac{c.T}{||c.T||}\cdot\frac{e.T}{||e.T||}$$

The factors *c.T* and *e.T* are represented in the standard vector space model, where each component corresponds to a term in the vocabulary. This results in a weighted sum of the number of common words in the query context and article context.

Measuring the similarity between contexts in this way has one major drawback: if alternative terms for one meaning are used, the similarity will be low even if the contexts denote the same entity.

6.2 Ranking with Aggregated Semantic Information

In this approach, the baseline cosine similarity is combined with additional information derived from a topic model over Wikipedia articles. This information holds the documents probability distribution over all topics from which the three topics with highest probability are used as additional features for both the query and the article text. To account for the divergence of query and article topic distributions, the symmetric Kullback-Leibler divergence of them is added as a dedicated feature. This is given by

$$\begin{aligned} D_{sym}(q,p) &= \frac{1}{2}\left(D(q,p)+D(p,q)\right)\\ D(q,p) &= \sum_{i=1}^{N} p(t_i)\log\left(\frac{p(t_i)}{q(t_i)}\right), \end{aligned}$$

where N is the number of topics in the topic model, and $p(t_i)$ is the probability of topic i in the document.

It should be noted, that the query text is in general much shorter than the article text and hence the computed topic distribution is less representative for the query text than for the article text.

Hence, the overall feature vector consists of

$$\begin{aligned} \Phi(c,e) &= \left[\phi_{cos}|\phi_{c.\mathcal{T}}|\phi_{e.\mathcal{T}}|\phi_{D_{sym}}\right]\\ \phi_{cos} &= \cos(c.T,e.T)\\ \phi_{c.\mathcal{T}} &= P(t_i|c.T), \text{ for the 3 most probable topics in } c.T\\ \phi_{e.\mathcal{T}} &= P(t_i|e.T), \text{ for the 3 most probable topics in } e.T\\ \phi_{D_{sym}} &= D_{sym}(c.\mathcal{T},e.\mathcal{T}). \end{aligned}$$

6.3 Ranking with Aggregated Semantic Information and Weighted Context Information

When context information is condensed into only a few singular measures, the ranking model has no chance to judge the actual influence of a specific word. Therefore we augmented the information presented to the model by the weighted context information, i.e. each common word is given as a feature whose value is its $tf \times idf$ score in the article text. Instead of a binary representation, additional importance is given to words that are important in the candidate entities article and hence potentially indicative.

We additionally add

$$\phi_{tfidf} = \begin{cases} \widetilde{w}_{i,e}, & \forall w_i \in e.T \cap c.T \\ 0, & \text{else} \end{cases} \tag{5}$$

where $\widetilde{w}_{i,e}$ denotes the $tf \times idf$ score of the i-th common word in the article text of e, to the overall feature vector.

Detecting Non-Listed Entities

Many entities appear in text, that are not present in Wikipedia and should hence not be related to a Wikipedia entity. To evaluate the models ability to detect if an entity is not present in Wikipedia a scenario was created, that simulates this. Here, not all the possible candidates were presented to the model, instead a given fraction was deliberately left out to present entities not mentioned in Wikipedia (non-listed entities e_{out}).

The feature vectors in this scenario are built of the same feature set as described above plus the additional feature

$$\phi_{out} = \mathbb{1}(e, e_{out}).$$

6.4 Ranking with Word-Topic Correlation

An alternative representation is to model the word-topic correlation. This is being done by correlating each common word with the complete topic distribution of the article. This representation is similar to the word-category correlation employed by [Bunescu and Pasca, 2006], with the difference that categories are substituted with topics and additionally the representation is not binary but involves the probability of each specific topic to be present in the article. The intuition is that this way, the most descriptive feature vectors can be build using

$$\phi_{topic_{w,t}} = P(t_i|e.T), \text{ if } w \in c.T \cap e.T \quad \forall i = 1, ..., 200.$$

For each of the common words $w \in c.T \cap e.T$, these vectors contain a group of features (i.e. the distribution of topics). This relates each word distinctly to the topic distribution extracted from the article text, which is a much more expressive summary than the other approaches discussed above.

7 Training and Testing

For our ongoing work we restricted the experiments to persons and considered the 3207 name phrases (first and last names) which correspond to at least 2 entities in German Wikipedia. In this set there are on average 2.5 entities per name phrase. In further experiments we will include all persons as well as other named entities like locations and organizations.

For training we use links in Wikipedia as names phrases n and extract the corresponding context $c(n)$ from the neighborhood of n. On average each person article has 12.1 links to other articles in Wikipedia. We randomly split each of these context sets into training and test queries, such that 90% are used for training and 10% for testing.

We used the ranking SVM described (3) with the associated inequalities (4). As feature function $\Phi(c_k, d_i)$ we used a linear kernel.

8 Results

We first evaluated the approach using cosine similarity and aggregated information inferred from the application of the above mentioned topic model. Since this information could potentially change with the width of the extracted context, we created two scenarios. In the first scenario, the context window is taken above the 25 left and 25 right neighboring words of the name in the query, in the second scenario above the 50 left and 50 right neighboring words. Both context representations naturally include the name itself. The article of the true entity is given as positive example and represented as a context of the first 50 resp. 100 words in the article, also assuming that the most descriptive and distinctive information is given in this snapshot. It could be shown, that enlarging the context does not increase the models performance, as was also observed by [Gooi and Allan, 2004] in their approach to cross-document coreference resolution, hence this and the other presented results are acquired on a 50 word context. The best performance was achieved as shown in table 5.

	Training	Test
F_{micro}	85.88	83.03
F_{macro}	87.34	78.96

Table 5: Performance (in %): Ranking SVM using cosine similarity, most probable topics and symmetric Kullback-Leibler divergence.

While the best result for the simple cosine similarity approach was a macro F-measure of 75.54%, the usage of semantic information improves the performance by 3.42% to an F-measure of 78.96%.

We then evaluated the approach using additional weighted context information. The results presented in table 6 show that again performance can be increased from 78.96% to now 84.85%, which motivates the approach using a more complex combination of context and topic associtation.

	Training	Test
F_{micro}	91.11	87.99
F_{macro}	91.83	84.85

Table 6: Performance (in %): Ranking SVM using cosine similarity, most probable topics, symmetric Kullback-Leibler divergence and weighted context information.

As described above, context information is important for the correct disambiguation of entities. In the following experiment, this is represented as a correlation between common words and the topics associated with the candidate entities article text as described in 6.4. Due to constraints in the implementation, it was not possible to produce results on the complete set of ambiguous names in time but only on a reduced set.

A reduced dataset was created over 500 ambiguous names yielding 1072 entities to be disambiguated and a sufficiently high number of training (5441) and test instances (1072). Context and training parameters were taken from the previous experiments. As table 7 shows, there is a dramatical increase in performance compared to the previous approaches to an F-measure of 97.01%, i.e. only 32 of the presented entities were not disambiguated correctly. To assess the models ability to deal with non-listed entities, 10% of the entities were simulated to be non-listed. This results in 970 listed and 102 non-listed entities, from which all were correctly marked as such. The number of false associations among Wikipedia entities is reduced to 14, which is due to the fact, that the remaining entities were simulated as non-listed. As table 8 shows, the performance was not reduced but shows equally good results. The even slightly increased F-measure is due to the fact, that some of the previously incorrectly disambiguated entities were now by chance in the set of non-listed entities.

	Training	Test
F_{micro}	99.17	97.01
F_{macro}	98.91	96.05
P_{macro}	99.09	95.57
R_{macro}	99.04	97.01

Table 7: Performance (in %): Ranking SVM using word-topic correlation.

	Training	Test
F_{micro}	99.23	97.76
F_{macro}	99.00	96.70
P_{macro}	99.19	96.29
R_{macro}	99.13	97.53

Table 8: Performance (in %): Ranking SVM using word-topic correlation with non-listed entities.

Since all non-listed entities are marked correctly, the micro performance has equal precision and recall values, which could not be observed in the other approaches, where the model was not able to rank all non-listed entities correctly. Although this dataset is considerably smaller than the ones in previous experiments, the results look very promising and should also apply to larger corpora.

We additionally evaluated a variant of the approach in [Bunescu and Pasca, 2006]: Instead of using only the top-level Wikipedia categories, the correlation between common words and all Wikipedia categories assigned to an article is used for the ranking approach. In order to keep the experiment comparable to the word-topic correlation approach, the data set is the same. Table 9 shows, that the performance is increased by 1.59 points to an F-measure of 98.6%.

	Training	Test
F_{micro}	99.65	98.6
F_{macro}	99.49	98.1
P_{macro}	99.55	97.9
R_{macro}	99.57	98.6

Table 9: Performance (in %): Ranking SVM using word-category correlation

[Bunescu and Pasca, 2006] reduced the number of treated categories, with the effect that more persons share categories and hence the categories are less distinctive, which can be a reason for the lower performance of only 84.8% accuracy as compared to 98.6% that were achieved here.

[Bunescu and Pasca, 2006] did not restrict the regarded context to the common words in query and article. But in fact, the usage of only common words to model context similarity is already rather restrictive. If they are additionally presented in pairs with categories, a nearly perfect model is achievable.

One of them is due to the ratio of entities and categories. For the complete corpus of 198903 Wikipedia person articles exist 16201 categories. Neglecting the 3996 categories that hold year of birth (1758) and year of death (2238) information, 12205 categories remain. From these, 2377 affect only one person. Hence, the number of categories is rather small in relation to the number of persons.

Note that result achieved with the word-topic correlation feature is only 1.59 points lower, but the topic model was only allowed to have 200 topics where as in the word-category correlation feature more than 4000 categories are used.

Although the topic correlation feature yielded slightly worse results than the category correlation feature, improved results are likely to be achieved using a better trained topic model or hierarchical topic models [Blei *et al.*, 2004] that can better incorporate category information. Wikipedia offers a good dataset for the evaluation of such models but generally their training can be performed on nearly any dataset.

Moreover, the uniqueness of entity pages was not always guaranteed in the version of Wikipedia used in this thesis. The ambiguous surface name *Jens Jessen* was mapped onto the three entities *Jens Jessen*, *Jens Jessen (Ökonom)* and *Jens Jessen (Journalist)*, where actually the entities *Jens Jessen* and *Jens Jessen (Ökonom)* were one and the same entity, although presented in two articles. A slightly different formulation of the disambiguation model could perform a consistency check on the uniqueness of entity pages, from which the model itself can only gain.

The most crucial assumption, e.g. that of correct links, was also proven not to hold generally. A false link results in an error in the disambiguation model, that at the current stage was identifiable only through manual analysis. This analysis showed for example that the human annotators mixed the two entities denoted by the name *John Barber*, e.g. the inventor of the gas turbine and an English race driver, whereas the disambiguation model identified them correctly.

9 Summary and Conclusion

This paper describes a model, that correctly assigns textual mentions of entities in German text to their representation in an external knowledge resource, e.g. Wikipedia. A ranking SVM was used with a diverse set of feature representations, that use simple cosine similarity, weighted context representation and sophisticated topics as well as the Wikipedia category set.

The presented results are comparable to those achieved on English datasets and can compete with those achieved on similar datasets with comparable ambiguities in names.

It is shown, that the approach may be used for the disambiguation of German named entities and is extendable to the more general task of concept disambiguation.

We have demonstrated, that it is not necessary to rely on categories manually assigned to the Wikipedia articles, but instead the application of topic models as an replacement for these categories achieves equally good results. This has the positive effect, that the system can be translated to any collection descrbing named entities, that is not endowed with categorization. In this way time-consuming annotations of this type can be omitted.

A challenging question is, how the disambiguation of one entity affects the disambiguation of other entities or related concepts mentioned for example in the same document. This could be investigated using for example a cascade of communicating disambiguation models.

10 Acknowledgement

The work presented here was funded by the German Federal Ministry of Economy and Technology (BMWi) under the THESEUS project.

References

[Bagga and Baldwin, 1998] Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proc. of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 79–85, San Francisco, California, 1998.

[Bekkerman and McCallum, 2005] Ron Bekkerman and Andrew McCallum. Disambiguating web appearances of people in a social network. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 463–470, New York, NY, USA, 2005. ACM.

[Bhattacharya and Getoor, 2005] Indrajit Bhattacharya and Lise Getoor. Relational clustering for multitype entity resolution. In *Proc. Fourth International Workshop on MultiRelational Data Mining (MRDM2005)*, 2005.

[Blei *et al.*, 2003] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[Blei *et al.*, 2004] David M. Blei, Michael I. Jordan, Thomas L. Griffiths, and Joshua B Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Proc. NIPS Advances in Neural Information Processing Systems 16*, 2004.

[Bunescu and Pasca, 2006] Razvan C. Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proc. of EACL*, pages 9–16, 2006.

[Cucerzan, 2007] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proc. 2007 Joint Conference on EMNLP and CNLL*, pages 708–716, 2007.

[Gooi and Allan, 2004] Chung H. Gooi and James Allan. Cross-document coreference on a large scale corpus. In *HLT-NAACL*, pages 9–16, 2004.

[Han *et al.*, 2004] Hui Han, C. Lee Giles, Hongyuan Zha, Cheng Li, and Kostas Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 296–305, New York, NY, USA, 2004. ACM.

[Hassel *et al.*, 2006] J. Hassel, B. Aleman-Meza, and I. B. Arpinar. Ontology-driven automatic entity disambiguation in unstructured text. pages 44–57, 2006.

[Huang *et al.*, 2006] Jian Huang, Seyda Ertekin, and C. Lee Giles. Efficient name disambiguation for large-scale databases. In *Proceedings of PKDD*, pages 536–544. PKDD, 2006.

[Joachims, 2002] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002)*, 2002.

[Kendall, 1955] Maurice Kendall. *Rank Correlation Methods.* Hafner, 1955.

[Mann and Yarowsky, 2003] Gideon S. Mann and David Yarowsky. Unsupervised personal name disambiguation. In *Proc. of the seventh conference on Natural language learning at HLT-NAACL 2003*, volume 4, pages 33–40, 2003.

[Miller and Charles, 1991] G.A. Miller and W.G. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):128,, 1991.

[Milne, 2007] David Milne. Computing semantic relatedness using wikipedia link structure. In *Proc. of the New Zealand Computer Science Research Student Conference (NZCSRSC'2007)*, 2007.

[Porter, 2001] Martin F. Porter. Snowball: A language for stemming algorithms.
http://snowball.tartarus.org, 2001.

[Sang and Meulder, 2003] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 142–147, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[Waltinger and Mehler, 2008] Ulli Waltinger and Alexander Mehler. Who is it? context sensitive named entity and instance recognition by means of wikipedia. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence (WI-2008)*, 2008.

[Wentland *et al.*, 2008] Wolodja Wentland, Johannes Knopp, Carina Silberer, and Matthias Hartung. Building a multilingual lexical resource for named entity disambiguation, translation and transliteration. In *Proc. of the Sixth International Language Resources and Evaluation (LREC'08)*, 2008.

[Wikimedia, 2009] Wikimedia.
http://stats.wikimedia.org/de/tablesrecenttrends.htm.
Retrieved on Feb. 22. 2009.

Condensed Random Sets for Efficient Quantitative Modelling of Gene Annotation Data

Frank Rügheimer
Biologie Systémique, Institut Pasteur
75015 Paris, France
frueghei@pasteur.fr

Ernesto William De Luca
Otto-von-Guericke University of Magdeburg
39106 Magdeburg
ernesto.deluca@ovgu.de

Abstract

With the widespread use of annotations in biological databases efficient models for statistical properties of set-valued attributes become increasingly relevant. In this work we introduce condensed random sets (CRS) as compact representations of distributions over annotation sets. The approach is discussed for both unorganized term vocabularies and term hierarchies, applied to an annotated yeast genome dataset and evaluated in comparison to an alternative representation. Encouraged by the results of the evaluation we explore further applications by pointing out how the representation can be used to support the construction of new semantic similarity measures for information retrieval.

1 Introduction

Genome sequencing has become a widely used tool in modern biology. Yet, in higher organisms, genomic information alone does not suffice to predict and characterize the manner in which particular gene products affect metabolic and signaling pathways, as it does not reflect the large number of interactions in the cell. To obtain a better understanding of biological processes the investigation of other layers of the cell machinery that are closer to the biological function has recently drawn much attention, e.g. the regulatory mechanisms involving RNAs [Toledo-Arana *et al.*, 2009] (transcriptome) and eventually the resulting proteins and their post-translational modifications themselves (proteome).

Due to a combination of recent advances of experimental techniques and extensive efforts to systematically survey literature, biologists have succeeded in establishing curated collections of information concerning gene products and their role for a number of model organisms. One of the results of those efforts was the realization that the same genes are frequently involved in several, sometimes seemingly unrelated biological processes. That information has been released to the public in the form of standardized public databases in which gene identifiers are associated with annotations terms describing their function. Using associated relational knowledge representations such as the Gene Ontology, annotation terms can be organized and linked with each other. In particular the Gene Ontology defines a hierarchy of annotation terms thus allowing to specify properties on different levels of detail.

In the present work we concern ourselves with the enrichment of relational knowledge representations with quantitative information extracted from annotated reference datasets. In particular we have investigated sets of annotation terms on the biological processes linked to the products of known genes. Observed statistical relationships between those annotations, and between terms and their generalizations were overlaid with term hierarchies extracted from the Gene Ontology. The resulting data representation can be used, e.g.:

1. To summarize and compare properties of datasets;
2. To compute likely expansions of coarse annotations to a higher level of detail, e.g. when making predictions from incomplete information or integrating data from different measurements;
3. To improve semantic similarity measures by taking into account empirically derived statistical relationships between annotation terms.

In the following section we establish how datasets featuring annotations with multiple terms can be modelled using condensed random sets. Following that we extend that approach to integrate it with relational knowledge representations in the form of term hierarchies. Both variants of the model are then applied to and evaluated on an annotated dataset for the bakers and brewers yeast *Saccharomyces cerevisiae*, which has been extensively studied as a eukaryotic model organism (Section 4). The results are compared to a representation based on an independent modelling of annotation terms. Finally Section 5 explores the prospect of applying the modeled term-set distributions for the construction of new context-specific semantic similarity measures.

2 Condensed Random Sets

Due to their relative flexibility and extensibility annotations have become a popular way to enrich existing data. Unlike conventional attributes that may only take one value out of a fixed domain, the same data-object may be simultaneously annotated with several terms that together describe a property. Denoting the set of potentially admissible annotation terms as Ω, annotations instantiate a set-valued attribute A^* that takes values from the set 2^Ω formed by the subsets of Ω. Apart from the simple list of associated terms, that very information may yield other interesting findings. For the annotated yeast genome, for example, it allows to investigate whether the activity of a gene is specific to a biological process or not. Conversely, when analysing expression data, the interpretation of the process annotations yields lists of candidates for pathways or functions that are affected by targeted interventions. Applied to whole genomes, the focus shifts from individual sets to frequency distributions over sets. From these distributions, one can obtain a quantitative characterization, for instance, of the level of complexity (fraction of genome involved) and relative importance (fraction of specialized genes/proteins) of biological processes in the organisms of interest.

In the probabilistic framework such a distribution over the possible annotation sets gives rise to a *Random Set* [Nguyen, 1978]. The two characteristics suggested as complexity and specificity assessments respectively correspond to the one-point coverage and the single-element probabilities of the random set, with a distribution p^* specifying the probability of each individual annotation-term combination. Since the number of combinations grows exponentially with the number of admissible annotation terms, however, a direct representation strategy allows for a very limited choice of annotation terms only. Even if all values can be represented in memory, providing estimates for a large number of – in most cases very small – probabilities with acceptable precision would require unrealistically large samples [Wasserman, 2006].

Fortunately, many applications do not require representations with detailed probability values for all set-valued outcomes. Due to their role in interpretations probabilities of singletons and the probability of term coverage by set-valued annotations provide useful information summaries. By focusing on these pieces of information the condensed random sets achieve a compact representation of statistical information regarding set attributes.

The condensed random set approach [Rügheimer, 2007], builds on a partitioning of the set of the subsets of a sample space Ω and a mapping of set-distributions to a probability/possibility distribution over the condensed domains. In the formalization of that approach a special attribute value is introduced to label outcomes that are multi-valued w.r.t. a frame of discernment Ω or correspond to the empty set. For simplicity, the representation is initially discussed for the case of an unstructured repository of annotation terms.

Definition 1 *Let Ω be a set of distinct labels. Furthermore let $\omega^\diamond$ be a special symbol uniquely associated with and not already contained in Ω. Consider a mapping σ from the set of subsets 2^Ω to the* **extended set universe** $\Omega \cup \{\omega^\diamond\}$

$$\begin{aligned} \sigma:\ 2^\Omega &\ \rightarrow\ \Omega \cup \{\omega^\diamond\} \\ \forall S \subseteq \Omega:\ \sigma(S) &\ =\ \begin{cases} \omega & \textit{if } S = \{\omega\}, \omega \in \Omega, \\ \omega^\diamond & \textit{otherwise.} \end{cases} \end{aligned} \tag{1}$$

We call σ the **set reduction mapping** *w.r.t. Ω.*

It is easily seen that σ preserves the distinction between singleton elements of 2^Ω, but collects the multi-valued outcomes in a separate class. Consider now a set of objects or cases O and their description via a set-valued attribute A^* taking values from 2^Ω. Using the definition of the set reduction mapping, it is possible to define a condensed set-valued attribute $A^\diamond$ that is linked to the values of A^*:

Definition 2 *Let A^* be a set-valued attribute $A^* : O \rightarrow 2^\Omega$. Additionally let $\sigma : 2^\Omega \rightarrow \Omega \cup \{\omega^\diamond\}$ denote the set reduction mapping w.r.t. Ω. The* **condensed set-valued attribute $A^\diamond$ induced by** A^* *is a mapping:*

$$\begin{aligned} A^\diamond:\ O &\ \rightarrow\ \Omega \cup \{\omega^\diamond\} \\ \forall o \in O:\ o &\ \mapsto\ \sigma(A^*(o)). \end{aligned} \tag{2}$$

The relation between the attribute domain conveyed by the set reduction mapping is illustrated in Figure 1. The underlying term set Ω is referred to as the *basic domain* of the condensed set-valued attribute $A^\diamond$ (written $\Omega = \mathrm{bdom}(A^\diamond)$).

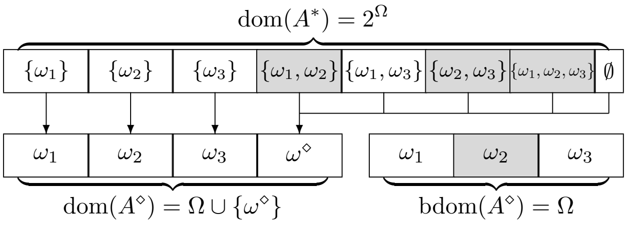


Figure 1: Domains of a set-valued attribute A^*, the induced condensed set-valued attribute $A^\diamond$ and underlying basic domain Ω. Arrows indicate the set reduction mapping w.r.t Ω. Shaded elements of $\mathrm{dom}(A^*)$ mark multi-valued outcomes covering ω_2.

Per Definition 2 the values of $A^\diamond$ depend directly on the values of A^*. Consequently a probability distribution p^* over $\mathrm{dom}(A^*)$ induces a probability distribution $p^\diamond$ over $\mathrm{dom}(A^\diamond)$, which summarizes p^*.

$$\begin{aligned} p^\diamond(\omega) &= P^*(\{S\colon \sigma(S) = \omega\}) \\ &= P^*(\sigma^{-1}(\omega)) \\ &= \begin{cases} p^*(\{\omega\}) & \text{if } \omega \in \mathrm{bdom}(A^\diamond), \\ \sum\limits_{\substack{S \in \mathrm{dom}(A^*) \\ |S| \neq 1}} p^*(S) & \text{if } \omega = \omega^\diamond. \end{cases} \end{aligned} \tag{3}$$

It is important to realize that for any element $\omega \in \mathrm{bdom}(A^\diamond)$, the value $p^\diamond(\omega)$ refers to the probability of ω being the *only* element in an annotation list, rather that just being one of them. The probability mass originally associated with multi-valued outcomes $S\colon S \in \mathrm{dom}(A^*), |S| > 1$ or with the empty-set outcome is assigned to a surrogate attribute value $\omega^\diamond$ in the condensed probability distribution. This approach has two immediate benefits: Since $p^\diamond$ is still a probability distribution, well established operations of the probabilistic framework like conditioning and marginalization can be employed with this representation. In addition to that, Definition 2 can be applied to estimate the condensed probability distributions directly from data, that is without prior computation of the distribution p^*.

Since they represent the non-ambiguous cases, singleton annotations are enriched in many real-world datasets. In the biological application considered here 56.9% of all genes are annotated with just one term. Should all annotations consist of a single term (no ambiguity) the representation is equivalent to a probability distribution over $\mathrm{dom}(A) = \mathrm{bdom}(A^\diamond) = \Omega$ as $p^\diamond(\omega) = p(\omega)$ and $p^\diamond(\omega^\diamond) = 0$ hold for that case. To support the reconstruction of one-point coverages, however, a richer representation is required. Given a probability distribution p^* for a set-valued attribute A^* taking values from 2^Ω, the one-point coverage of individual elements $\omega \in \Omega$ is computed as follows:

$$\begin{aligned} \forall \omega \in \Omega : \operatorname{opc}(\omega) &= P^*(S : S \subseteq \Omega \wedge \omega \in S) \\ &= \textstyle\sum_{\substack{S \subseteq \Omega \\ \omega \in S}} p^*(S). \end{aligned} \tag{4}$$

For each $\omega \in \Omega$ one element of the sum in the right-hand expression of Equation 4 is obtained directly from the distribution $p^\diamond$ of the induced condensed attribute $A^\diamond$. For $S = \{\omega\}$ the summand is recovered due to the equality $p^*(S) = p^*(\{\omega\}) = p^\diamond(\omega)$. To represent the contribution from all other subsets of Ω, the latter are encoded as proportions relative to $p^\diamond(\omega^\diamond)$ (called coverage factors):

Definition 3 *Let p^* denote a distribution linked to a set-valued attribute (A^*) over 2^Ω and $p^\diamond$ the distribution over the domain $\operatorname{dom}(A^\diamond)$ of an induced condensed set-valued attribute $A^\diamond$ obtained by applying equation 3. Then the* **coverage function** *$c^\diamond$ relative to multi-valued outcomes of A^* is defined as a function*

$$\begin{aligned} c^\diamond : \Omega &\rightarrow [0,1] \\ \omega &\mapsto \begin{cases} \frac{\sum_{\substack{S \subseteq \Omega, \omega \in S, \\ |S|>1}} p^*(S)}{p^\diamond(\omega^\diamond)} & \textit{if } p^\diamond(\omega^\diamond) > 0, \\ 1 & \textit{otherwise.} \end{cases} \end{aligned} \tag{5}$$

For $p^\diamond(\omega^\diamond)$ the value $c^\diamond(\omega)$ denotes the conditional probability for ω being *contained* in a non-singleton outcome. Although the contributions to the one-point coverage could have been stored directly, the representation via relative coverage factors was chosen to better support probabilistic conditioning and marginalization operations. In the case $p^\diamond(\omega^\diamond) = 0$, the conditional coverage factors are undefined, but can be set to a constant. Alternatively the problem can be avoided altogether by using a Laplace correction.

Like the distribution $p^\diamond$, the *relative coverage factors* assigned by $c^\diamond$ can be computed directly from data. Replacing the sum in Equation 4 the one-point coverage may now be rewritten as

$$\begin{aligned} \forall \omega \in \Omega : \operatorname{opc}(\omega) &= \textstyle\sum_{\substack{S \subseteq \Omega \\ \omega \in S}} p^*(S) \\ &= p^\diamond(\omega) + p^\diamond(\omega^\diamond) \cdot c^\diamond(\omega) \end{aligned} \tag{6}$$

In the following, the term *condensed distribution* is understood to refer to a tuple $(p^\diamond, c^\diamond)$ that is formed by a condensed probability distribution and the corresponding coverage function.

The advantage of the condensed set-valued attribute $A^\diamond$ and the function $p^\diamond$ and $c^\diamond$ as compared to the full random set representation is the reduction of the number of parameters. For each term of the attribute domain only the probability for the singleton outcome and the coverage factor need to be stored. For practical reasons, it is also advantageous to explicitly represent the combined probability mass of all multi-valued outcomes, which is required for the calculation of every one-point coverage. This raises the total number of model parameters to $2|\Omega| + 1$. With the condensed random sets the number of distribution parameters grows linearly in the size of the underlying base domain Ω. In contrast, a full distribution over sets would have to encode the probabilities of $2^{|\Omega|}$ possible instantiations.

3 Application to Term Hierarchies

So far the individual annotation terms were considered as largely unrelated. In practise, however, terms are frequently organized in a hierarchy. For several application fields such term hierarchies are specified as part of an ontology. We refer to such term relations using the functions parent_H/$\operatorname{children}_H$ to denote a terms direct predecessors/descendants in the hierarchy and more generally anc_H/desc_H for compatible terms of different specificity. The hierarchical term structure acknowledges that annotations may originate from different sources and provide information on distinct levels of detail. This means that it is no longer sufficient to trace which terms or labels have been used in an annotations itself, but also to consider other applicable terms that are implied. For example, in the hierarchy depicted in Figure 2 the term a_1 is a generalization of a_{12}. Therefore, whenever a_{12} applies to a situation, so does a_1. In contrast, if a case is labeled only as a_1 we do not know which of the more specific labels a_{11}, a_{12}, a_{13} apply (Figure 2). However, the probability of different refinement alternatives may be estimated by looking at the conditional distributions for term usage in fine grained annotations in reference data or simply the remainder of the dataset.

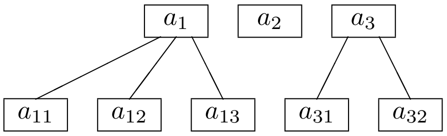


Figure 2: Attribute Value Hierarchy Example

3.1 Model Construction

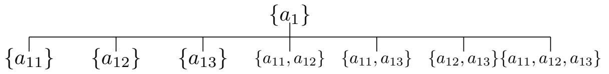


Figure 3: Possible Refinements of Label a_1 in the Hierarchy from Figure 2

A general approach to use the condensed random set framework to deal with to such hierarchies has been described in [Rügheimer and Kruse, 2008]. Each branch in the term hierarchy H is associated with a condensed random set that models the empirical distributions of the possible expanded annotations in reference data. The combined set of labels in the term hierarchy is denoted by $\mathcal{L}$.

The above representation strategy presumes that the expanded annotations w.r.t. different parent labels are (statistically) independent of one another given those parents in the hierarchy. Applied to all expandable labels of the hierarchy, this leads to the data structure depicted in Figure 4. For each non-leaf label λ_r an additional label $\lambda_r^\diamond$ is introduced. In the condensed representation the conditional probability assigned to that label refers to the event that the label λ_r is split into more than one applicable child labels during the next refinement step. This is complemented with conditional coverage factors, which are stored for each element in the direct refinement of λ_r.

To estimate the model parameters from empirical data the Equations 3 and 5 are applied to the branch distributions of non-leaf labels λ_r. The respective reference set is formed by those observations, for which λ_r is both applicable and has been expanded on the observed frame. In

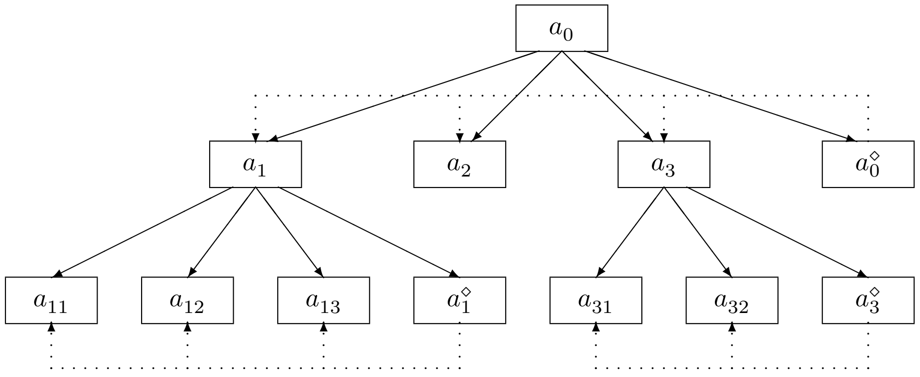


Figure 4: Extended attribute value hierarchy as data structure for the condensed representation of distributions over multi-valued instantiations (conditional probabilities and coverage factors indicated by solid and dotted arrows respectively)

that case information on the applicability of the individual child labels of λ_r is available too. An algorithm to calculate the branch distributions for a given label hierarchy H is given below (Figure 5). For each instantiation from the training data, all compatible nodes in the term hierarchy are marked. Following that, affected branch distributions are traversed to update counters for element coverage (in the case of a multi-label instantiation) or for the occurrence of the respective singleton. Counter updates for the first label in each instantiation are delayed until a distinction of single- and multi- label instantiations becomes possible. After all instantiations have been processed the condensed distribution function and coverage functions are calculated.

The branch distribution on the originally set-valued selections of applicable labels from the elementary refinement of λ_r is represented using the condensed distribution $p^\diamond_{H,\lambda_r}$, with the new element $\lambda_r^\diamond$ representing the non-singleton annotation sets, and the associated coverage function $c^\diamond_{H,\lambda_r}$. The lcorr parameter denotes a user-defined constant for an optional Laplace correction, which is applied for both the induction of branch probabilities and conditional coverage factors (the latter being instances of a two class problem). The bounding of the normalization factors ensures that all marginal probabilities will be defined, even if the Laplace correction is not applied. This guarantee does not extend to conditional branch probabilities though. By altering the normalization factors the algorithm is easily adapted to alternative interpretations of the non-expanded values in the training data set.

3.2 Recalling Information

To facilitate the use of the above representation to model distributions, let us now address how stored information is accessed. To recover a set-distribution from an existing model, the conditional branch distributions on the hierarchy are recombined into respective distributions on the frames. For singleton outcomes this amounts to multiplying branch probabilities along a path of label refinement, i.e. $\forall \lambda \in \mathcal{L}$:

$$p^{*\prime}_H(\{\lambda\}) = \prod_{\lambda' \in (\{\lambda\} \cup \mathrm{anc}_H(\lambda)) \setminus \lambda_0} p^\diamond_{H,\mathrm{parent}_H(\lambda')}(\lambda'). \quad (7)$$

In general the approximation will be imperfect. In addition to the unavoidable sampling error, the branch distributions do not distinguish between real singletons and cases where a label is merely the only applicable element in the local branch. Provided sufficient training data is available,

Algorithm 1 Inducing Branch Distributions

```
procedure getFrequencies(H,Observations,lcorr)
    ResetCounters
    for currentObs $\in$ Observations do
        for $\lambda \in$ currentObs do
            markLabel($H, \lambda$)
            markAncestors($H, \lambda$)
        end for
        updateCounters($H$)
        obsCnt $\leftarrow$ obsCnt $+ 1$
    end for
    $\lambda_r \leftarrow$ rootLabel($H$)
    while validLabel($\lambda_r$) do
        nrm_p $\leftarrow \max\{1,$ GetNumSgltExp($\lambda_r$) $+$ GetNumMltvExp($\lambda_r$)
                    $+(1 + |\mathrm{children}_H(H, \lambda_r)|) \cdot \mathrm{lcorr}\}$
        nrm_c $\leftarrow \max\{1,$ GetNumMltvExp($\lambda_r$) $+ 2 \cdot \mathrm{lcorr}\}$
        for $\lambda \in$ childrenH($\lambda_r$) do
            $p^\diamond_{H,\lambda_r}(\lambda) \leftarrow \frac{(\mathrm{GetNumAsSglt}(\lambda) + \mathrm{lcorr})}{\mathrm{nrm\_p}}$
            $c^\diamond_{H,\lambda_r}(\lambda) \leftarrow \frac{(\mathrm{GetNumAsCvrd}(\lambda) + \mathrm{lcorr})}{\mathrm{nrm\_c}}$
        end for
        $p^\diamond_{H,\lambda_r}(\lambda_r^\diamond) \leftarrow \frac{(\mathrm{GetNumMltvExp}(\lambda_r) + \mathrm{lcorr})}{\mathrm{nrm\_p}}$
        $\lambda_r \leftarrow$ nextLabelInDepthFirstSearchOrder($H$)
    end while
end procedure
```

Algorithm 2 Inducing Condensed Branch Distributions (Counting)

```
procedure updateCounters(H)
    $p \leftarrow$ rootLabel($H$)
    while validLabel($p$) do
        nMarkedChildren $\leftarrow 0$
        for $c \in$ ChildrenH($p$) do
            if isMarked(c) then
                if nMarkedChildren $= 0$ then
                    firstChild $\leftarrow c$
                    nMarkedchildren $\leftarrow 1$
                else
                    countAsCovered($c$)
                    nMarkedChildren $\leftarrow$ nMarkedChildren $+ 1$
                end if
            end if
        end for
        if nMarkedChildren $= 1$ then
            countAsSingleton(firstChild)
            countSgltExpansion($p$)
        else if nMarkedChildren $> 1$ then
            countAsCovered(firstChild)
            countMltvExpansion($p$)
        end if
        clearMark($H, p$)
        $p \leftarrow$ nextMarkedLabelInDepthFirstSearchOrder($H$)
    end while
end procedure
```

Figure 5: Calculation of Condensed Branch Distributions from Data

a higher precision can be obtained by adding a separate set of branch distributions though.

The one point coverages of individual labels are retrieved by recursively accumulating conditional probabilities and coverage factors for each elementary refinement leading to the label in question. For a single recursion step the reconstructed one-point coverage of a given label is obtained by application of Equation 6. Because each branch distribution refers only to those cases where the respective ancestor labels are applicable, the result is than multiplied with the respective one-point coverage for the ancestors: $\forall \lambda \in \mathcal{L} \neq \lambda_0,\ \lambda_r \overset{\text{def}}{=} \text{parent}_H(\lambda):$

$$\begin{aligned} \text{opc}'_H(\lambda) \ &= \ \text{opc}'_H(\lambda_r) \cdot \Big(p^{\diamond}_{H,\lambda_r}(\lambda) \\ &\quad + \ p^{\diamond}_{H,\lambda_r}(\lambda^{\diamond}_r) \cdot c_{H,\lambda_r}(\lambda) \Big), \end{aligned} \tag{8}$$

where λ_r is used as a shorthand notation for the parent label of λ in the hierarchy and $\lambda^{\diamond}_r$ the corresponding surrogate label that indicates multiple applicable elements in the extension of λ_r. For each level in the hierarchy an additional factor is supplied until the root label λ_0 is reached. If the empty annotation sets are excluded the one-point coverage of that label is always one[1]. To efficiently compute one-point coverages for several elements of a frame an implementation would reuse partial results whenever the recursion runs over shared ancestors in the hierarchy. Under the assumption that applicability of the individual labels within an elementary refinement is independent for non-singleton instantiations, the one-point coverages can also be used to approximate probability values for annotations sets with more than one term, though the approximation quality is lower than for the singletons.

Finally case-specific information on one-point coverages and probabilities can be integrated to allow reasoning. This is achieved by temporarily fixing conditional branch distributions to externally supplied inputs. In the next step the distributions on the target frames are recomputed with the provided values taking precedence over those supplied by the model. Recursions are broken early whenever one of the externally provided values is encountered and only the missing conditional branch probabilities are supplemented by the model.

4 Experimental Evaluation

The evaluation has been conducted on an annotated genome dataset released to the public via the Saccharomyces Genome Database project [SGD Curators, a]. The SGD-project maintains a curated database that summarizes published results about the function of the genes and gene products of the baker's and brewer's yeast Saccharomyces cerevisiae, as well as their respective roles in biological processes and their intracellular activity sites. Annotation follows a domain-wide standard defined in the gene-ontology [The Gene Ontology Consortium, 2000]. The latter also defines term relations that allow to link annotations on different levels of specificity to each other. The terms are organized into three non-overlapping term hierarchies for the tree aspects of annotation (processes, functions, cellular component). Each of these term hierarchies forms a separate branch of the ontology and is connected to the other two only via the common root node.

Since the full annotation is very detailed, a considerable fraction of the annotation terms is only applied to a very small subset of the database. Due to their extremely low term coverage it is not well-justified to include them into a statistical analysis. To provide a standardized broader view of the represented knowledge, less specific versions of the ontology have been released by the consortium. These so-called "slim ontologies" define species-specific subsets of comparatively general Gene Ontology terms and are usually released together with the full annotation data collected in coordinated efforts to analyze the genome and proteome of selected model organisms. The dataset used in the experiments was based on a projection of the full SGD annotations to a subset of relatively broad gene-ontology terms – the GO-Slim terms for yeast [SGD Curators, b]. Term that were not included in the in the slim version of the ontology were mapped to their most specific hierarchical ancestor in the reduced term set. Both that mapping and the GO-Slim itself are maintained at the SGD website.

To evaluate the proposed framework, test its underlying assumptions and compare its predictions with those of alternative frameworks, we implemented three different approaches:

- A model in which presence or absence of elements in a set are encoded using binary variables. The latter variables are treated as independent, so the distribution of set-instantiations is obtained as a product of binary distributions for the state of the elements of the underlying carrier set. The set-distribution is described via its one-point-coverage.
- A condensed distribution model using an unstructured attribute domain
- An enriched term hierarchy using condensed random sets for the representation of branch distribution (described in Section 3).

For the experiment all models were trained using the distribution of annotation sets from a randomly sampled subsets of the yeast genome. The resulting distribution models were then compared with the distribution of the annotation term combinations on the remaining genes. To that end approximation quality and generalization were evaluated using several measures that emphasize either overall quality of fit, the representation of singleton outcomes or the prediction of element coverage. To increase robustness of that evaluation against sampling effects a cross-validation strategy was employed for all experiments.

4.1 Data Preparation and Experimental Setup

Due to the structure of the SGD projects internal database, each assignment of an annotation term to a gene is represented as separate database record. Apart form the gene name and annotation term these records contain supplementary information, such as alternative gene names, the annotation aspect class, types of information sources used to assign the annotation, references to the location of the gene within the genome or connected publications.

Historically several genes have been described and named by two or more research group independently. Often these groups investigated seemingly unrelated biological functions in different organisms. Only later, when refined sequencing and sequence comparison techniques allowed to locate genes within a genome and identify homologue genes in different species, these discoveries have

[1] Otherwise, empty instantiations can easily be represented by inserting a "virtual" root label with an unnormalized branch distribution at the top of the hierarchy. In that case, the one point coverage of the original root label is computed using Equation 8, whereas the one-point coverage of the new root label is set to one.

been found to refer to identical or analogous objects. As a result several genes are known by more than just one name. In order ensure that annotations can be attributed correctly, the first step of preprocessing consisted in mapping all alternative gene names to unique standard identifiers which are used throughout all subsequent processes.

Following that, the records where filtered w.r.t. the annotation aspect given. For the purpose of this evaluation the annotation w.r.t. the "biological process" aspect was chosen. In comparison to the other annotation classes, the annotations on the biological processes provide a comparatively reliable and extensive higher-level description of the role of the gene product in the organism. In the remaining part of the database, annotations for individual genes are still spread over several database records. To better support a gene-based view on the data annotations where grouped by the genes they refer to. The resulting file summarizes the known biological function for each of 6849 genes using 909 distinct annotation sets.

In parallel, the preprocessing routines assembled information about the annotation scheme employed. To that end the term hierarchy structure was extracted from the ontology and converted them into a domain specification for the hierarchical version of the condensed distribution models. Similar domain specifications were prepared for the non-hierarchical version and for the model based on independent binary variables. In those cases however the domain specifications were limited to a list of annotation terms, that is the information on term organization was disregarded. The generated domain specifications were later used to preconfigure distribution models in the training phase.

The above preprocessing method resulted in a database of annotation sets for 6849 genes. To study the properties of the model types this set was split into five partitions with genes randomly assigned (4 partition with 1370 genes each and one partition with 1369 genes). To limit sampling effects, the evaluation measures were computed in a 5-fold cross-validation process [Kohavi, 1995] with a different partition serving as a test data set and the remaining partitions providing training data in each run.

4.2 Parameter Estimation

Using the model configuration files prepared in the preprocessing step and the training data for each validation run, the different model types were trained for the distribution of gene annotation sets. In the case of the reference model with independent binary variables the parameter set consists of one value per element in the carrier set, which describes the probability of an instantiation containing that very element. The modeled probability $\hat{P}(S)$ of any set-instantiation $S \subseteq \Omega$ is obtained by computing the products

$$\hat{P}(S) = \left(\prod_{\omega \in S} \mathrm{opc}(\omega) \right) \cdot \left(\prod_{\omega \in \Omega \setminus S} (1 - \mathrm{opc}(\omega)) \right), \quad (9)$$

with the model parameters $\mathrm{opc}(\omega)$ denoting the (estimated) probability of the element ω being an element of the realization. Coverage rates for elements in the carrier set are estimated from the frequencies of the two possible outcomes "element is present in the instantiation" and "element is absent in the instantiation".

For the condensed distribution and hierarchy-based condensed distribution model the parameters are singleton probabilities and conditional coverage factors either for the distribution as a whole, or – in the hierarchical version – for subtrees of the label hierarchy. For a detailed description of parameters and the model induction procedures see Sections 2 and 3 respectively. In all cases, the parameters were estimated from the observed frequencies in the training data with a Laplace correction of 0.5 applied.

4.3 Evaluation Measures

Having discussed the different model classes, their training and the general evaluation method, we shall now investigate the evaluation measures employed for this task. The measures where chosen to provide complementary information on how well different aspects of the set-distribution are captured by each model type.

Log-Likelihood To describe those measures it is assumed that all models are evaluated against a test data set $D_{\mathrm{tst}} = (d_1, d_2, \ldots, d_m)$ with each d_i formed by the set of annotations applicable to one particular gene. A common way to evaluate the fit of a probability-based model M is to consider the likelihood of the observed test data D_{tst} under the model, that is, the conditional probability estimate $\hat{P}(D_{\mathrm{tst}} \mid M)$. The closer the agreement between test data and model, the higher that likelihood will be. The likelihood is also useful to test model generalization, as models that overfit the training data tend to predict low likelihoods for test datasets drawn from the same background distribution as the training data. To circumvent technical limitations concerning the representation of and operations with small numbers in the computer, the actual measure used in practice is based on the logarithm of the likelihood:

$$\begin{aligned} \log L(D_{\mathrm{tst}}) &= \log \prod_{d \in D_{\mathrm{tst}}} P(d \mid M) && (10) \\ &= \sum_{d \in D_{\mathrm{tst}}} \log P(d \mid M). && (11) \end{aligned}$$

In that formula the particular term used to estimate the probabilities $P(d \mid M)$ of the records in D are model-dependent. Since the likelihood takes values form $[0, 1]$ the values for the log-transformed measure are from $(-\infty, 0]$ with larger values (closer to 0) indicating better fit. The idea of the measure is that the individual cases (genes) in both the training and the test set are considered as independently sampled instantiations of a multi-valued random variable drawn from the same distribution. The Likelihood of a particular test database of size m is computed as the product of the likelihoods of its m records. Due to the low likelihood of individual sample realizations even for good model approximation, the Log-Likelihood is almost always implemented using the formula given in Equation 11, which yields intermediate results within the bounds of standard floating point format number representations.

One particular difficulty connected with the Log-Likelihood, resides in the treatment of previously unobserved cases in the test data set. If such values are assigned a likelihood of zero by the model then this assignment entails that the whole database is considered as impossible and the Log-Likelihood becomes undefined. In the experiment this undesired behavior was countered by applying a Laplace correction of $\mathrm{lcorr} = 0.5$ during the training phase. This modification ensures that all conceivable events that have not been covered in the training data are modeled with a small non-zero probability estimate and allow the resulting measures to discriminate between databases containing such records.

Average Record Log-Likelihood: The main idea of the log-likelihoods measure is to separately evaluate the likelihood of each record in the test database with respect to the model and consider the database construction process a sequence of a finite number of independent trials. As a result log-likelihoods obtained on test databases of different sizes are difficult to compare. By correcting for the size of the test database one obtains an average record log-likelihood as a more suitable measure:

$$\mathrm{arLL}(D_{\mathrm{tst}}) = \frac{\log L(D_{\mathrm{tst}})}{|D_{\mathrm{tst}}|} \tag{12}$$

Note that in the untransformed domain the mean of the log-likelihoods corresponds to the geometric mean of the likelihoods, and is thus consistent with the construction of the measure from a product of evaluations of independently generated instantiations.

Singleton and Coverage Rate Errors: In addition to the overall fit between model and data, it is desirable to characterize how well particular properties of a set-distribution are represented. In particular it has been pointed out that the condensed distribution emphasizes the approximation of both singleton probabilities and the values of the element coverage. To assess the quality of the approximations from an application-oriented viewpoint and compare it to results achieved using other methods, two additional measures – $\mathrm{d}_{\mathrm{sglt}}$ and $\mathrm{d}_{\mathrm{cov}}$ – have been employed. These measures are based on the sum of squared errors for the respective values over all elements of the base domain:

$$\mathrm{d}_{\mathrm{sglt}} = \sum_{\omega\in\Omega} \left(p'(\omega) - p(\omega)\right)^2, \tag{13}$$

$$\mathrm{d}_{\mathrm{cov}} = \sum_{\omega\in\Omega} \left(\mathrm{opc}'(\omega) - \mathrm{opc}(\omega)\right)^2. \tag{14}$$

4.4 Experimental Results

For increased robustness of the results the evaluation was conducted using 5-fold cross-validation. In each of the five runs the models were trained using a Laplace correction of 0.5. To obtain a basis for the assessment and comparison of the different methods, the evaluation results of the individual runs were collected and – with the exception of the logL measure[2] – averaged. These results are summarized in the Tables 1–3.

$\log L$	arLL	$\mathrm{d}_{\mathrm{sglt}}$	$\mathrm{d}_{\mathrm{cov}}$
-9039.60	-6.60	0.067856	0.001324
-8957.19	-6.54	0.064273	0.001524
-9132.09	-6.67	0.060619	0.001851
-8935.82	-6.52	0.074337	0.001906
-9193.44	-6.72	0.059949	0.001321
	-6.61	0.065406	0.001585

Table 1: Evaluation Results for Model Using Independent Binary Variables (One-Point-Coverage) with Laplace Correction of 0.5

As anticipated the two condensed random set-based models achieve a considerably better fit to the test data (higher value of arLL-measure) than the model assuming independence of term coverages. Among the two CRS-based models the variant that uses the term hierarchy structure clearly benefits from this additional information and consistently yields better results than its competitor. The large error obtained for the prediction of singleton annotations in the model based on independent binary variables, points out the inadequacy of the independence assumption in the latter representation. In contrast, with their separate representation of singleton annotation sets, the CRS-based models show only small prediction errors for the singleton frequencies, though the incomplete separations between real singletons and single elements in local branch distributions appears to leads to a slightly increased error for the hierarchical version. This is consistent with the higher error d_{cov} of that model in the prediction of coverage factors. The two non-hierarchical models represent one-point coverages directly and therefore achieve identical prediction error[3].

$\log L$	arLL	$\mathrm{d}_{\mathrm{sglt}}$	$\mathrm{d}_{\mathrm{cov}}$
-7629.66	-5.57	0.000539	0.008293
-7559.38	-5.52	0.000457	0.011652
-7752.21	-5.66	0.000857	0.006998
-7529.83	-5.50	0.001014	0.004767
-7828.44	-5.72	0.000567	0.009961
	-5.59	0.000686	0.008334

Table 2: Evaluation Results for Condensed Distribution on Hierarchically Structured Domain with Laplace Correction of 0.5

$\log L$	arLL	$\mathrm{d}_{\mathrm{sglt}}$	$\mathrm{d}_{\mathrm{cov}}$
-7992.76	-5.83	0.000241	0.001342
-7885.19	-5.76	0.000222	0.001531
-8045.31	-5.87	0.000411	0.001838
-7839.16	-5.72	0.000612	0.001895
-8195.49	-5.99	0.000268	0.001316
	-5.83	0.00035	0.001584

Table 3: Evaluation Results for Condensed Distribution on Unstructured Domain with Laplace Correction of 0.5

5 Relevance for Semantic Similarity Measures

Measures of semantic similarity between concepts have been successfully applied in linguistics, where they are used in Word Sense Disambiguation [Patwardhan *et al.*, 2003], and in bioinformatics, where they are used to in connection with annotation databases to evaluate or enhance clustering or classification algorithms. The Gene Ontology [The Gene Ontology Consortium, 2000] has been developed with the aim of supporting users in using their biological background knowledge to find information in biological databases. But to actually retrieve the desired information it is necessary to relate the users query to stored pieces of information (e.g. documents). The majority of current search engines try to interpret the meaning of the query based on the keywords contained in it. The system uses these keywords to rank results by their degree of similarity to the applied query as defined by a similarity measure. If the keywords are well chosen, these methods fre-

[2] See the discussion on the arLL measure to review the argument why averaging Log-Likelihoods is not meaningful here

[3] The minor differences between the tables are merely artifacts of the two-factor decomposition of coverage factor in the condensed distribution.

quently provide an appropriate list of results. However, if the search terms are ambiguous or are used in different domains, then a rather inhomogeneous collection of results is returned. As this is the case for a considerable number of queries retrieval performance can be improved by applying automatic categorization / filtering techniques to separate those cases. In the biocomputing and the biomedical field semantic similarity measures have been employed to improve document retrieval [Lord *et al.*, 2003], [Pedersen *et al.*, 2007].

Whereas earlier semantic similarity measures were based on graph distance in a term hierarchy some of the more recent variants rely on measures of statistical interaction between pairs of terms [Lin, 1998] and context vectors, which essentially compare relative term coverage between the query and each semantic class [Patwardhan *et al.*, 2003]. In [De Luca, 2008], semantic prototype vectors were constructed from a combination of observed data and extensions acquired from ontological resources.

With condensed random sets it would be possible to obtain a more accurate representation of the distribution of annotation sets. Due to the additional parameters for modelling single valued annotations a typically large fraction of many real world datasets is represented with increased precision. Moreover term interaction are implicitly considered in the hierarchical version of the model. Currently the likelihood measures used in Section 4 are being developed into normalized similarity measures. Already the likelihood based assessment of similarity allow comparisons between groups of annotated objects as well as between groups and individual annotation sets. It can thus be applied both to compare clusters/groups (comparison: distribution–distribution) and to solve classification problems such as word sense disambiguation (comparison: instantiation–distribution).

6 Conclusions

Condensed Random Sets allow to efficiently model probability distributions over annotation sets. Because the number of model parameters is linear in the cardinality of the annotation term set, it can be applied to datasets that are inaccessible to a full random set representation.

It was demonstrated that the assumptions made to achieve this compact representation are in agreement with properties of a relevant real-world biological dataset leading to a high approximation quality – when compared to a reference approach with independent modeling of term annotations. At the same time the condensed representation allows to reduce the problem of overfitting, which constitutes another common problem with full random set representations.

A hierarchical version allows to condition distributions and to supplement information given on different levels of detail. Although the example discussed in this paper refers to a specific problem of biological data analysis, the internal representation employed is general enough to be applied to other random-set based knowledge models in a large field of applications.

References

[De Luca, 2008] Ernesto William De Luca. *Semantic Support in Multilingual Text Retrieval.* Shaker Verlag, Aachen, Germany, 2008.

[Kohavi, 1995] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proc. of the 14th Int. Joint Conference on Artificial Intellligence (IJCAI 95)*, pages 1137–1145, 1995.

[Lin, 1998] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning, Madison, WI*, pages 296–304, Madison, WI, USA, 1998.

[Lord *et al.*, 2003] P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble. Semantic similarity measures as tools for exploring the gene ontology. *Pac Symp Biocomput*, pages 601–612, 2003.

[Nguyen, 1978] Hung T. Nguyen. On random sets and belief functions. *Journal Math. Anal. Appl.*, 65:531–542, 1978.

[Patwardhan *et al.*, 2003] S. Patwardhan, S. Banerjee, and T. Pedersen. Using Measures of Semantic Relatedness for Word Sense Disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pages 241–257, Mexico City, Mexico, February 2003.

[Pedersen *et al.*, 2007] Ted Pedersen, Serguei V. S. Pakhomov, Siddharth Patwardhan, and Christopher G. Chute. Measures of semantic similarity and relatedness in the biomedical domain. *J. of Biomedical Informatics*, 40(3):288–299, 2007.

[Rügheimer and Kruse, 2008] Frank Rügheimer and Rudolf Kruse. An uncertainty representation for set-valued attributes with hierarchical domains. In *Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2008), Málaga, Spain*, 2008.

[Rügheimer, 2007] Frank Rügheimer. A condensed representation for distributions over set-valued attributes. In *Proc. 17. Workshop Computational Intelligence*, Karlsruhe, Germany, 2007. Universitätsverlag Karlsruhe.

[SGD Curators, a] SGD Curators. Saccharomyces genome database. (accessed 2008/11/16).

[SGD Curators, b] SGD Curators. SGD yeast gene annotation dataset (slim ontology version). via Saccharomyces Genome Database Project [SGD Curators, a]. (accessed 2008/11/16).

[The Gene Ontology Consortium, 2000] The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

[Toledo-Arana *et al.*, 2009] Alejandro Toledo-Arana, Olivier Dussurget, Georgios Nikitas, Nina Sesto, Hélène Guet-Revillet, Damien Balestrino, Edmund Loh, Jonas Gripenland, Teresa Tiensuu, Karolis Vaitkevicius, Mathieu Barthelemy, Massimo Vergassola, Marie-Anne Nahori, Guillaume Soubigou, Béatrice Régnault, Jean-Yves Coppée, Marc Lecuit, Jörgen Johansson, and Pascale Cossart. The listeria transcriptional landscape from saprophytism to virulence. *Nature*, May 2009.

[Wasserman, 2006] Larry Wasserman. *All of Nonparametric Statistics.* Springer Texts in Statistics. Springer, New York, 2006.

An Exploitative Monte-Carlo Poker Agent (*Resubmission*[*])

Immanuel Schweizer, Kamill Panitzek, Sang-Hyeun Park and Johannes Fürnkranz
TU Darmstadt - Knowledge Engineering Group,
Hochschulstr. 10 - Darmstadt - Germany

Abstract

We describe the poker agent AKI-REALBOT which participated in the 6-player Limit Competition of the third Annual AAAI Computer Poker Challenge in 2008. It finished in second place, its performance being mostly due to its superior ability to exploit weaker bots. This paper describes the architecture of the program and the Monte-Carlo decision tree-based decision engine that was used to make the bot's decision. It will focus the attention on the modifications which made the bot successful in exploiting weaker bots.

1 Introduction

Poker is a challenging game for AI research because of a variety of reasons [Billings *et al.*, 2002]. A poker agent has to be able to deal with *imperfect* (it does not see all cards) and *uncertain* information (the immediate success of its decisions depends on random card deals), and has to operate in a *multi-agent* environment (the number of players may vary). Moreover, it is not sufficient to be able to play an optimal strategy (in the game-theoretic sense), but a successful poker agent has to be able to exploit the weaknesses of the opponents. Even if a game-theoretical optimal solution to a game is known, a system that has the capability to model its opponent's behavior may obtain a higher reward. Consider, for example, the simple game of *rock-paper-scissors* aka *RoShamBo* [Billings, 2000], where the optimal strategy is to randomly select one of the three possible moves. If both players follow this strategy, neither player can gain by unilaterally deviating from it (i.e., the strategy is a *Nash equilibrium*). However, against a player that always plays *rock*, a player that is able to adapt its strategy to always playing *paper* can maximize his reward, while a player that sticks with the "optimal" random strategy will still only win one third of the games. Similarly, a good poker player has to be able to recognize weaknesses of the opponents and be able to exploit them by adapting its own play. This is also known as *opponent modeling*.

In every game, also called a *hand*, of fixed limit Texas Hold'em Poker, there exist four game states. At the *pre-flop* state, every player receives two *hole cards*, which are hidden to the other players. At the *flop*, *turn* and *river* states, three, one and one *community cards* are dealt face up respectively, which are shared by all players. Each state ends with a *betting* round. At the end of a hand (the *showdown*) the winner is determined by forming the strongest possible five-card poker hand from the players's hole cards and the community cards. Each game begins by putting two forced bets (*small blind* and *big blind*) into the pot, where the big blind is the minimal betting amount, in this context also called *small bet* (SB). In pre-flop and flop the betting amount is restricted to SBs, whereas on turn and river one has to place a *big bet* (2× SB). At every turn, a player can either *fold*, *check/call* or *bet/raise*.

In this paper, we will succinctly describe the architecture of the AKI-REALBOT poker playing engine (for more details, cf. [Schweizer *et al.*, 2009]), which finished second in the AAAI-08 Computer Poker Challenge in the 6-player limit variant. Even though it lost against the third and fourth-ranked player, it made this up by winning more from the fifth and sixth ranked player than any other player in the competition.

2 Decision Engine

2.1 Monte-Carlo Search

The Monte Carlo method [Metropolis and Ulam, 1949] is a commonly used approach in different scientific fields. It was successfully used to build AI agents for the games of bridge [Ginsberg, 1999], backgammon [Tesauro, 1995] and Go [Bouzy, 2003]. In the context of game playing, its key idea is that instead of trying to completely search a given game tree, which is typically infeasible, one draws random samples at all possible choice nodes. This is fast and can be repeated sufficiently frequently so that the average over these random samples converges to a good evaluation of the starting game state.

Monte-Carlo search may be viewed as an orthogonal approach to the use of evaluation functions. In the latter case, the intractability of exhaustive search is dealt with by limiting the search depth and the use of an evaluation function at the leaf nodes, whereas Monte Carlo search deals with this problem by limiting the search breadth at each node and the use of random choice functions at the decision nodes. A key advantage of Monte Carlo search is that it can deal with many aspects of the game without the need for explicitly representing the knowledge. Especially for poker, these include *hand strength*, *hand potential*, *betting strategy*, *bluffing*, *unpredictability* and *opponent modeling* [Billings *et al.*, 1999]. These concepts, for which most are hard to model explicitly, are considered implicitly by the outcome of the simulation process.

In each game state, there are typically three possible actions, *fold*, *call* and *raise*.[1] AKI-REALBOT uses the sim-

[*] This paper is a resubmission and has been accepted at the 32nd Annual Conference on Artificial Intelligence (KI 2009).

[1] The AAAI rules restrict the number of bets to four, so that a

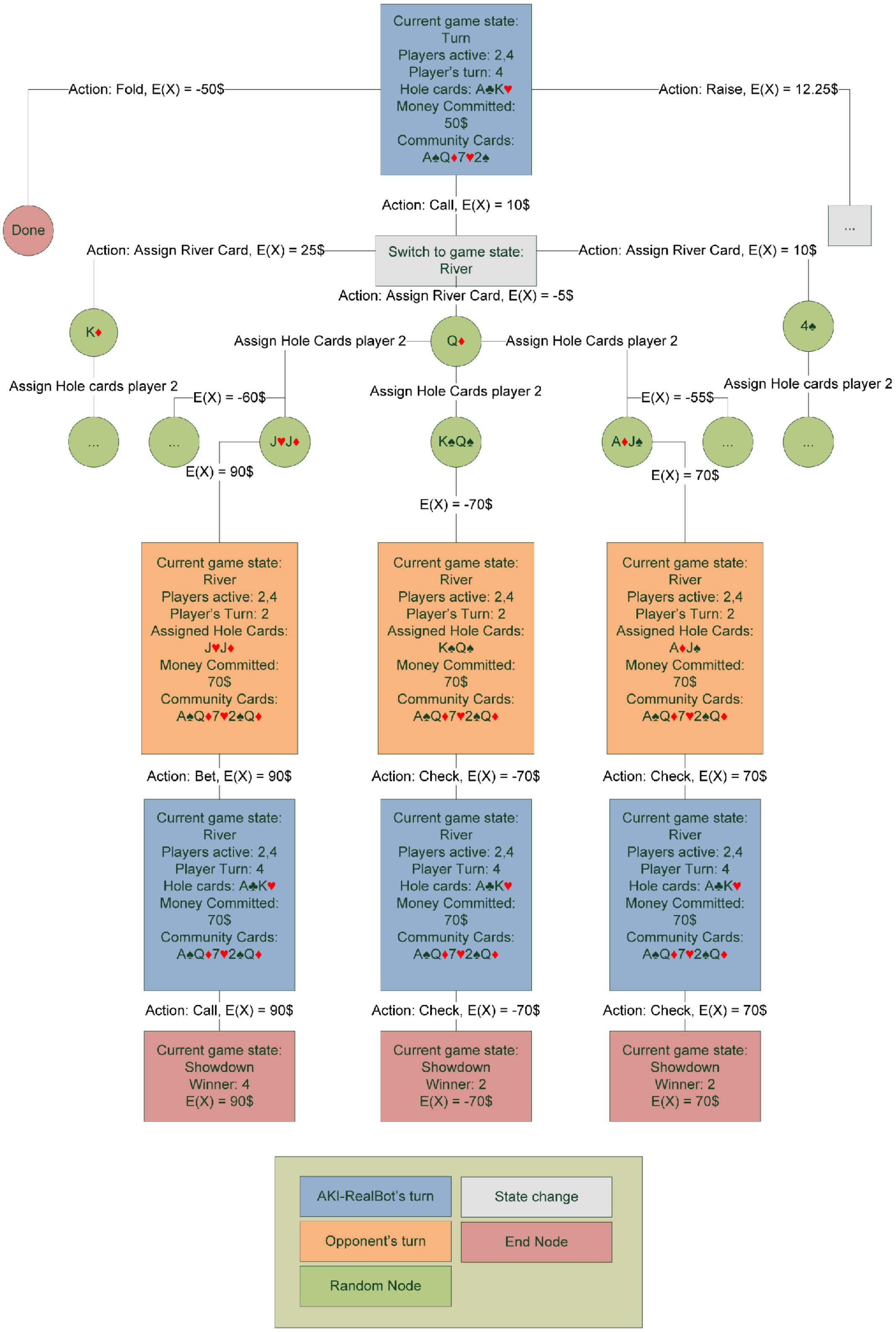


Figure 1: Monte Carlo Simulation: the figure depicts an example situation on the turn, where AKI-REALBOT is next to act (top). The edges represent in general the actions of players or that of the chance player. For the decisions *call* or *raise* (middle and right path), two parallel simulations are initiated. The path for the *call* decision for example (in the middle), simulates random games until the showdown (the river card is Qd, the opponent cards are estimated as KsQs, and both players check on the river.) and the estimated loss of 70$ is backpropagated along the path.

ulated expected values (EV) for them to evaluate a decision. These EVs are estimated by applying two independent Monte-Carlo searches, one for the call action and the other one for the raise action (cf. Figure 1). Folding does not have to be simulated, since the outcome can be calculated immediately. Then at some point, these search processes are stopped by the Time Management component [Schweizer *et al.*, 2009], which tries to utilize the available time as effective as possible. Since an increase in the number of simulated games also increases the quality of the EVs and therefore improves the quality of the decision, a multi-threading approach was implemented.

Our Monte-Carlo search is not based on a uniformly distributed random space but the probability distribution is biased by the previous actions of a player. For this purpose, AKI-RealBot collects statistics about each opponent's probabilities for folding (f), calling (c), and raising (r), thus building up a crude opponent model. This approach was first described in [Billings *et al.*, 1999] as selective sampling. For each played hand, every active opponent player is assigned hole cards. The selection of the hole cards is influenced by the opponent model because the actions a player takes reveal information about the strength of his cards, and should influence the sample of his hole cards. This selection is described in detail in Section 3. After selecting the hole cards, at each player's turn, a decision is selected for this player, according to a probability vector (f, c, r), which are estimated from the previously collected data.

Each community card that still has to be unveiled is also randomly picked whenever the corresponding game state change happens. Essentially the game is played to the showdown. The end node is then evaluated with the amount won or lost by AKI-RealBot, and this value is propagated back up through the tree. At every edge the average of all subtrees is calculated and represents the EV of that subtree. Thus, when the simulation process has terminated, the three decision edges coming from the root node hold the EV of that decision. In a random simulation, the better our hand is, the higher the EV will be. This is still true even if we select appropriate samples for the opponents' hole cards and decisions as long as the community cards are drawn uniformly distributed.

2.2 Decision Post-Processing

AKI-RealBot post-processes the decision computed by the Monte-Carlo search in order to increase the adaptation to different agents in a multiplayer scenario even further with the goal of exploiting every agent as much as possible (in contrast to [Billings *et al.*, 1999]). The exploitation of weak opponents is based on two simple considerations:

1. Weak players play too tight, i.e. they fold too often
2. Weak players play too loose (especially post-flop), which is the other extreme: they play too many marginal hands until the showdown

These simply defined weak players can be easily exploited by an overall aggressive play strategy. It is beneficial for both types of players. First, if they fold too often, one can often bring the opponent to fold a better hand. Second, against loose players, the hand strength of marginal hands increase, such that one can win bigger pots with them than usual. Besides the aggressive play, the considerations imply a loose strategy. By expecting that AKI-RealBot can *outplay* the opponent, it tries to play as many hands as possible against weaker opponents.

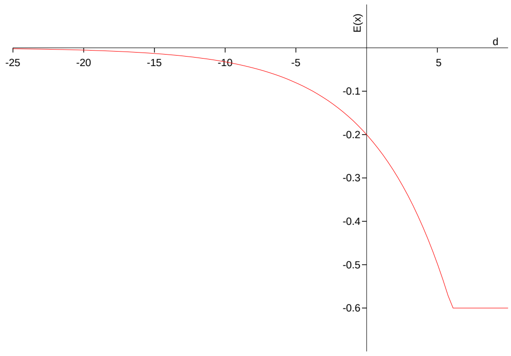


Figure 2: Aggressive Pre-flop Value $\delta(d)$

This kind of commonly known expert-knowledge was explicitly integrated. For this purpose, so-called *decision bounds* were imposed on the EVs given by the simulation. This means that for every opponent, AKI-RealBot calculates dynamic upper and lower bounds for the EV, which were used to alter the strategy to a more aggressive one against weaker opponents. $E(f)$ will now denote the EV for the *fold* path, while $E(c)$ and $E(r)$ will be the values for *call* and *raise* respectively. Without post-processing, AKI-RealBot would pick the decision x where $E(x) = \max_{i=\{f,c,r\}} E(i)$.

Aggressive Pre-flop Value

The lower bound is used for the pre-flop game state only. As long as the EV for folding is smaller than the EV for either calling or raising (i.e., $E(f) < \max(E(c), E(r))$), it makes sense to stay in the game. More aggressive players may even stay in the game if $E(f) - \delta < \max(E(c), E(r))$ for some value $\delta > 0$. If AKI-RealBot is facing a weak agent W it wants to exploit its weakness. This means that AKI-RealBot wants to play more hands against W. This can be achieved by setting $\delta > 0$. We assume that an agent W is weak if he has lost money against AKI-RealBot over a fixed period of rounds. For this purpose, AKI-RealBot maintains a statistic over the number of small bets (SB) d, that has been lost or won against W in the last $N = 500$ rounds. For example, if W on average loses 0.5 SB/hand to AKI-RealBot then $d = 0.5 \times 500 = 250$ SB. Typically, d is in the range of $[-100, 100]$. Then, the *aggressive pre-flop value* δ for every opponent is calculated as

$$\delta(d) = \max(-0.6, -0.2 \times (1.2)^d)$$

Note that $\delta(0) = -0.2$ (SB), and that the value of maximal aggressiveness is already reached with $d \approx 6$ (SB). That means, that AKI-RealBot already sacrifices in the initial status $d = 0$ some EV (maximal -0.2 SB) in the pre-flop state, in the hope to outweight this drawback by *outplaying* the opponent post-flop. Furthermore, if AKI-RealBot has won in the last 500 hands only more than 6 SB against the faced opponent, it reaches its maximal *optimism* by playing also hands which EVs were simulated as low as ≈ -0.6 SB. This makes AKI-RealBot a very aggressive player pre-flop, especially if we consider that δ for more than one active opponent is calculated as the average of their respective δ values.

raise is not always possible.

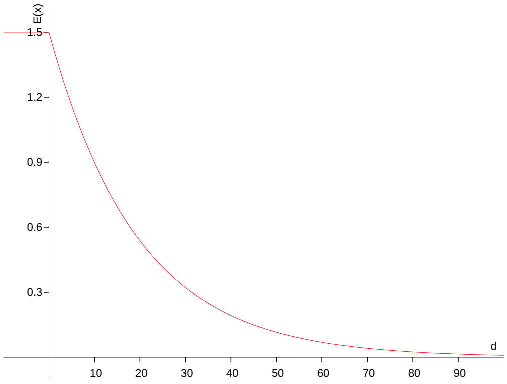


Figure 3: Aggressive Raise Value $\rho(d)$

Aggressive Raise Value

The upper bound is used in all game states and makes AKI-REALBOT aggressive on the other end of the scale. As soon as this upper bound is reached, it will force AKI-REALBOT to raise even if $E(c) > E(r)$. This will increase the amount of money that can be won if AKI-REALBOT is very confident about his hand strength. This upper bound is called the *aggressive raise value* ρ.

$$\rho(d) = \min(1.5, 1.5 \times (0.95)^d)$$

Here, the upper bound returns $\rho(0) = 1.5$ for the initial status $d = 0$, which is 1.5 times the SB and therefore a very confident EV. In fact, it is so confident that this is also the maximum value for ρ. The aggressive raise value is not influenced if we lose money against a player. If, on the other hand, AKI-REALBOT wins money against an agent W, it will slowly converge against zero, resulting in a more and more aggressive play.

As said before, the value of d is calculated based on a fixed amount of past rounds. It is therefore continuously changing with AKI-REALBOT's performance over the past rounds. The idea is to adapt dynamically to find an optimal strategy against any single player. On the other hand, it is easy to see that this makes AKI-REALBOT highly vulnerable against solid, strong agents.

3 Opponent Modeling

In general, the opponent modeling of the AKI-REALBOT, which is used to adjust the implemented Monte-Carlo simulation to the individual behavior of the other players, considers every opponent as a straight-forward player. That means, we assume that aggressive actions indicate a high hand strength and passive actions a low hand strength. Within the simulation, the opponent's hand strength is guessed based on the action he takes. So if a player often folds in the pre-flop phase but calls or even raises in one special game this means he has probably a strong hand. In addition the opponent modeling tries to map cards to the actions every player takes.

AKI-REALBOT has two different routines that enables it to guess hole cards according to the opponent model. Which routine is used depends on the game state.

Pre-Flop State:

In pre-flop, we assume that the actions of a player are only based on his hole cards. He is either confident enough to raise or to make a high call, whereas making a small call may indicate a lower confidence in his hand. A high call is indicated by committing more than a big bet. In either case his observed fold ratio f and call ratio c are used to calculate an upper and lower bound for the set of hole cards. It is common to divide the set of possible hole cards into *buckets*, where each bucket consists of hole cards of similar hand strength, to reduce the space of hole card combinations. We used five buckets, U_0 being the weakest bucket and U_4 (e.g. containing the cards AA) the strongest. The buckets have the following probability distribution: $p(U_0) = 0.65$, $p(U_1) = 0.14$, $p(U_2) = 0.11$, $p(U_3) = 0.07$, $p(U_4) = 0.03$.

In the first case of the above example, the upper bound U is set to the maximum possible bucket value ($U_h = 4$) because high confidence was shown. The lower bound is calculated by taking $l = c + f$ and relating this to the bucket. That means, the lower bound is set to exclude the hole cards, for which the player would only call or fold. If for example $f = 0.71$ and $c = 0.2$, the player raises only in 9% of cases. Since we assumed a straight-forward or honest player, we imply that he only does this with the top 9% of hole cards. So, the lower bound is set to U_3. Then, the hole cards for that player are selected randomly from the set of hole cards which lie between the bounds.

Post-Flop State

The second routine for guessing the opponents' hole cards is used when the game has already entered a post-flop state. The main difference is that the actions a player takes are now based on both hidden (his hole cards) and visible information (the board cards). Therefore, AKI-REALBOT has to estimate the opponent's strength also by taking the board cards into account. It estimates how much the opponent is influenced by the board cards. This is done by considering the number of folds for the game state flop. If a player is highly influenced by the board he will fold often on the flop and only play if his hand strength has increased with the board cards or if his starting hand was irrespectively very strong.

This information is used by AKI-REALBOT to assign hole cards in the post-flop game state. Two different methods are used here:

- *assignTopPair*: increases the strength of the hole cards by assigning the highest rank possible, i.e., if there is an ace on the board the method will assign an ace and a random second card to the opponent.
- *assignNutCard*: increases the strength of the hole cards even more by assigning the card that gives the highest possible poker hand using all community cards i.e. if there is again an ace on the board but also two tens the method will assign a ten and a random second card.

These methods are used for altering one of the player's hole card on the basis of his fold ratio f on the flop. We distinguish among three cases based on f, where probability values p_{Top} and p_{Nut} are computed.

$$\begin{array}{llcl}
(1) & f < \frac{1}{3} & \Rightarrow & p_{Top} = 3(f)^2 \in [0, \frac{1}{3}[\\
 & & & p_{Nut} = 0 \\
(2) & \frac{1}{3} \leq f < \frac{2}{3} & \Rightarrow & p_{Top} = \frac{1}{3} \\
 & & & p_{Nut} = \frac{1}{3}(3f-1)^2 \in [0, \frac{1}{3}[\\
(3) & f \geq \frac{2}{3} & \Rightarrow & p_{Top} = \frac{1}{3} \\
 & & & p_{Nut} = f - \frac{1}{3} \in [\frac{1}{3}, \frac{2}{3}]
\end{array}$$

Table 1: AAAI-08 Poker Competition Results: pairwise and overall performance of each entry

	Poki0	AKI-Real	DCU	CMURing	mcBot	GUS6
Poki0		65,176	2,655	18,687	29,267	214,840
AKI-RealBot	-65,176		-15,068	-2,769	30,243	348,925
DCU	-2,665	15,068		7,250	16,465	90,485
CMURing	-18,687	2,769	-7,250		7,549	92,453
mcBotUltra	-29,267	-30,243	-16,465	-7,549		16,067
GUS6	-214,840	-348,925	-90,485	-92,453	-16,067	
Total	330,822	296,293	126,657	76,848	-67,529	-763,091
avg. winnings/game	3934	3579	1512	939	-800	-9042
SB/Hand	0.656	0.588	0.251	0.152	-0.134	-1.514
Place	1.	2.	3.	4.	5.	6.

To be clear, *assignTopPair* is applied with a probability of p_{Top}, *assignNutCard* is applied with a probability of p_{Nut} and with a probability of $1-(p_{Top}+p_{Nut})$ the hole cards are not altered. As one can see in the formulas, the higher f is, the more likely it is that the opponent will be assigned a strong hand in relation to the board cards. Note that for both methods the second card is always assigned randomly. This will sometimes strongly underestimate the cards e.g. when there are three spade cards on the board *assignNutCard* will not assign two spade cards.

4 AAAI-08 Computer Poker Competition Results

AKI-RealBot participated in the 6-player Limit competition part of the Computer Poker Challenge at the AAAI-08 conference in Chicago. There were six entries: hyperborean08_ring aka Poki0 (University of Alberta), DCU (Dublin City University), CMURing (Carnegie Mellon University), GUS6 (Georgia State University), mcBotUltra and AKI-RealBot, two independent entries from TU Darmstadt.

Among these players, 84 matches were played with different seating permutations so that every bot could play in different positions. Since the number of participants were exactly 6, every bot was involved in all 84 matches. In turn, this yielded 504000 hands for every bot. In that way, a significant result set was created, where the final ranking was determined by the accumulated win/loss of each bot over all matches.[2]

Table 1 shows the results over all 84 matches. All bots are compared with each other and the win/loss statistics are shown in SBs. Here it becomes clear that AKI-RealBot exploits weaker bots because the weakest bot, GUS6, loses most of it's money to AKI-RealBot. Note, that GUS6 lost in average more than 1.5 SBs per hand, which is a worse outcome than by folding every hand, which results in an avg. loss of 0.25 SB/Hand. Although AKI-RealBot loses money to DCU and CMURing it manages to rank second, closely behind Poki0, because it is able to gain much higher winnings against the weaker players than any other player in this field. Thus, if GUS6 had not participated in this competition, AKI-RealBot's result would have been much worse.

5 Conclusion

In this paper, we have described the poker agent AKI-RealBot that finished second in the AAAI-08 Poker Competition. Its overall performance was very close to the winning entry, even though it has lost against three of its opponents in a direct comparison. The reason for its strong performance was its ability to exploit weaker opponents. In particular against the weakest entry, it won a much higher amount than any other player participating in the tournament. The key factor for this success was its very aggressive opponent modeling approach, due to the novel adaptive post-processing step, which allowed it to stay longer in the game against weaker opponents as recommended by the simulation.

Based on these results, one of the main further steps is to improve the performance of AKI-RealBot against stronger bots. An easy way would be to adopt the approaches of the strongest competitors of the competition, for which there exists a multitude of publications. But, we see also yet many possible improvements for our exploitative approach, which we elaborate in [Schweizer *et al.*, 2009] and are currently working on.

References

[Billings *et al.*, 1999] Darse Billings, Lourdes Peña Castillo, Jonathan Schaeffer, and Duane Szafron. Using probabilistic knowledge and simulation to play poker. In *AAAI/IAAI*, pages 697–703, 1999.

[Billings *et al.*, 2002] Darse Billings, Aaron Davidson, Jonathan Schaeffer, and Duane Szafron. The challenge of poker. *Artif. Intell.*, 134(1-2):201–240, 2002.

[Billings, 2000] Darse Billings. Thoughts on RoShamBo. *ICGA Journal*, 23:3–8, 2000.

[Bouzy, 2003] Bruno Bouzy. Associating domain-dependent knowledge and monte carlo approaches within a go program. In *Joint Conference on Information Sciences*, pages 505–508, 2003.

[Ginsberg, 1999] Matthew L. Ginsberg. Gib: Steps toward an expert-level bridge-playing program. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 584–589, 1999.

[Metropolis and Ulam, 1949] N. Metropolis and S. Ulam. The monte carlo method. *J. Amer. Stat. Assoc.*, 44:335–341, 1949.

[Schweizer *et al.*, 2009] Immanuel Schweizer, Kamill Panitzek, Sang-Hyeun Park, and Johannes Fürnkranz. An exploitative monte-carlo poker agent. Technical Report TUD-KE-2009-02, TU Darmstadt, KE Group, 2009.

[Tesauro, 1995] Gerald Tesauro. Temporal difference learning and td-gammon. *Commun. ACM*, 38(3):58–68, 1995.

[2] The official results can be found at `http://www.cs.ualberta.ca/~pokert/2008/results/`

Learning Pattern Tree Classifiers Using a Co-Evolutionary Algorithm

Robin Senge and Eyke Hüllermeier
Department of Mathematics and Computer Science
University of Marburg, Germany
{senge,eyke}@informatik.uni-marburg.de

Abstract

Pattern tree induction has recently been introduced as a novel method for classification. Roughly speaking, a pattern tree is a hierarchical, tree-like structure, whose inner nodes are marked with generalized (fuzzy) logical operators, and a pattern tree classifier consists of one such tree per class. Since a pattern tree can thus be considered as a kind of logical characterization of a class, the approach is very appealing from an interpretation point of view. Yet, as will be argued in this paper, the method that has originally been proposed for learning pattern trees is not optimal and offers scope for improvement. To overcome its disadvantages, we propose a new method which is based on the use of co-evolutionary algorithms. Experimentally, it will be shown that our approach is indeed able to outperform the original learning method in terms of predictive accuracy.

1 Introduction

Pattern tree induction has recently been introduced as a novel method for classification by Huang, Gedeon and Nikravesh [Huang *et al.*, 2008]. Roughly speaking, a pattern tree is a hierarchical, tree-like structure, whose inner nodes are marked with generalized (fuzzy) logical operators, and whose leaf nodes are assigned to input attributes. A node takes the values of its descendants as input, applies the respective operator, and submits the output to its predecessor. Thus, a pattern tree implements a recursive mapping producing outputs in $[0, 1]$. A pattern tree classifier consists of a set of pattern trees, one for each class. A query instance to be classified is submitted to each tree, and a prediction is made in favor of the class whose tree produces the highest output.

Pattern trees are interesting for several reasons, especially from an interpretation point of view. In fact, each tree can be considered as a kind of logical description of a class. Alternatively, in the context of *preference learning* [Hüllermeier *et al.*, 2008], a tree can be seen as a utility function: Each class corresponds to a choice alternative, and the one with the highest utility is selected.

Even though first experiments seem to suggest that pattern tree classifiers perform reasonably well in terms of predictive accuracy, the learning algorithm originally proposed in [Huang *et al.*, 2008] is arguably not optimal. In particular, as will be explained in more detail later on, it translates training examples into examples for each tree in a questionable way. Besides, it expands trees by using a simple greedy strategy and, therefore, is susceptible to local optima. In this paper, we therefore propose an alternative method for training pattern tree classifiers which is based on the idea of optimization via co-evolution.

The rest of the paper is structured as follows. Section 2 gives a short introduction to pattern tree induction. Section 3 is devoted to our novel method of co-evolutionary pattern tree learning. Experimental results are presented in Section 4. The paper end with some concluding remarks and an outlook on future work in Section 5.

2 Pattern Trees

In this section, we briefly describe pattern trees and the original learning algorithm; for technical details, we refer to [Huang *et al.*, 2008]. Subsequently, we discuss some deficiencies of this method and motivate an alternative approach.

2.1 Tree Structure and Components

We proceed from the common setting of supervised learning and assume an attribute-value representation of instances, which means that an instance is a vector

$$\boldsymbol{x} \in \mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_m \enspace ,$$

where $\mathcal{X}_i$ is the domain of the i-th attribute A_i. Each domain $\mathcal{X}_i$ is discretized by means of a fuzzy partition, that is, a set of fuzzy subsets $F_{i,j}$ of $\mathcal{X}_i$ such that $\sum_j F_{i,j}(x) > 0$ for all $x \in \mathcal{X}_i$ (recall that a fuzzy set F_{ij} is an $\mathcal{X}_i \rightarrow [0, 1]$ mapping). The F_{ij} are often associated with linguistic labels (such as "small" or "large") and then also referred to as *fuzzy terms*. Each instance is associated with a class label

$$y \in \mathcal{Y} = \{y_1, y_2, \ldots, y_k\} \enspace .$$

A training example is a tuple $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$.

Unlike decision trees, which assume an input at the root node and output a class prediction at each leaf, pattern trees process information in the reverse direction. The input of a pattern tree is entered at the leaf nodes. More specifically, a leaf node is labeled by an attribute A_i and a fuzzy subset F of the corresponding domain $\mathcal{X}_i$. Given an instance $\boldsymbol{x} = (x_1, \ldots, x_m) \in \mathcal{X}$ as an input, the node produces $F(x_i)$ as an output,

Name	T-Norm	Code
MIN	$min\{a, b\}$	MIN
Algebraic	ab	ALG
Lukasiewicz	$max\{a + b - 1, 0\}$	LUK
EINSTEIN	$\frac{ab}{2-(a+b-ab)}$	EIN
Name	T-CoNorm	Code
MAX	$max\{a, b\}$	MAX
Algebraic	$a + b - ab$	COALG
Lukasiewicz	$min\{a + b, 1\}$	COLUK
EINSTEIN	$\frac{a+b}{1+ab}$	COEIN

Table 1: Fuzzy Operators

that is, the degree to which x_i is in F. This degree of membership is then propagated to the parent node.

Internal nodes are labeled by generalized logical or arithmetic operators, including

- t-norms and t-conorms [Klement *et al.*, 2002],
- weighted and ordered weighted average [Schweizer and Sklar, 1983; Yager, 1988].

A t-norm is a generalized conjunction, namely a monotone, associative and commutative $[0, 1]^2 \to [0, 1]$ mapping with neutral element 1 and absorbing element 0. Likewise, a t-conorm is a generalized disjunction, namely a monotone, associative and commutative $[0, 1]^2 \to [0, 1]$ mapping with neutral element 0 and absorbing element 1. Some examples of t-norms and t-conorms are shown in Table 1.

An ordered weighted average (OWA) combination of k values $v_1 \dots v_k$ is defined by

$$\text{OWA}_w(v_1 \dots v_k) \overset{\text{df}}{=} \sum_{i=1}^{k} w_i \cdot v_{\tau(i)}, \tag{1}$$

where τ is a permutation of $\{1 \dots k\}$ such that $v_{\tau(1)} \leq_{\tau(2)} \leq \dots \leq v_{\tau(k)}$ and $w = (w_1 \dots w_k)$ is a weight vector satisfying $w_i \geq 0$ for $i = 1 \dots k$ and $\sum_{i=1}^{k} w_i = 1$. Thus, just like the normal weighted average (WA), an OWA operator is parameterized by a set of weights.

Note that for $k = 2$, (1) is simply a convex combination of the minimum and the maximum. In fact, the minimum and the maximum operator are obtained, respectively, as the two extreme cases of (1): $w_1 = 1$ yields $\text{OWA}_w(v_1 \dots v_k) = v_{\tau(1)} = \min(v_1 \dots v_k)$ and $w_k = 1$ gives $\text{OWA}_w(v_1 \dots v_k) = v_{\tau(k)} = \max(v_1 \dots v_k)$. Therefore, the class of OWA operators nicely "fills the gap" between the largest conjunctive combination, namely the minimum t-norm, and the smallest disjunctive combination, namely the maximum t-conorm.

The result of an evaluation of an internal node is again propagated to its parent and so forth. The output produced by a tree is the output of its root node. Fig. 1 shows some examples.

A pattern tree classifier consists of a set of pattern trees pt_i, $i = 1, 2, \dots, k$, one for each class. Given a new instance $\boldsymbol{x}$ to be classified, a prediction is made in favor of the class whose tree produces the highest score:

$$\widehat{y} = \arg\max_{y_i \in \mathcal{Y}} pt_i(\boldsymbol{x}) \tag{2}$$

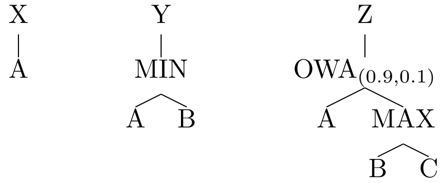


Figure 1: Pattern Tree Examples

2.2 Pattern Tree Induction

Following [Huang *et al.*, 2008], pattern trees are build one by one independently of each other. For each class, the induction method carries out the following steps:

1. Initialize with primitive pattern trees
2. Filter candidates by evaluation of their similarity to the class
3. Check stopping criterion
4. Recombine candidates using a set of fuzzy operators
5. Loop at step 2

During initialization, primitive pattern trees are created. They consist of only one leaf node. The first set of candidate trees is built by creating one such primitive pattern tree for each fuzzy term of each attribute.

In the second step, the "similarity" of each candidate tree with its respective class y_i is evaluated. Roughly speaking, this is done by comparing the subset A of examples of that class with the fuzzy subset B of instances as predicted by the tree: If $(\boldsymbol{x}, y)$ is a training example, then $A(\boldsymbol{x}) = 1$ if $y = y_i$ and $A(\boldsymbol{x}) = 0$ if $y \neq y_i$. Likewise, $B(\boldsymbol{x})$ is the output of the tree produced for the input $\boldsymbol{x}$. A and B are similar if high values $A(\boldsymbol{x})$ come along with high values $B(\boldsymbol{x})$ and vice versa. This can be quantified in terms of different measures of similarity, such as the generalized Jaccard coefficient

$$\frac{|A \cap B|}{|A \cup B|} = \sum_{\boldsymbol{x}_i} \frac{\min(A(\boldsymbol{x}_i), B(\boldsymbol{x}_i))}{\max(A(\boldsymbol{x}_i), B(\boldsymbol{x}_i))} .$$

Only candidate trees with a high degree of similarity are considered within the next steps. In step 4, every possible combination of two candidate trees, sticked together by one of the allowed fuzzy operators, is created (i.e., by taking the operator as a root and the two candidate trees as subtrees). All these candidates are used for the next iteration starting at step 2. The algorithm stops if a stopping criterion is satisfied, for example if the candidate trees reach a certain depth.

2.3 Discussion

The learning procedure as outlined above can be criticized for several reasons, notably the following. First, the learning algorithm implements a kind of greedy search in the hypothesis space. Since this space is extremely complex, it is likely to get stuck in local optima. Clearly, the complexity of the search space and the highly non-linear nature of the models prevents from the use of search algorithms which guarantee optimality. Yet, there is hope that better solutions can be found at the cost of an increased though still acceptable search effort. As mentioned previously, we shall resort to search methods from the field of evolutionary optimization.

Second, one may argue that the learning problem is made more difficult than necessary. In fact, as described above, the learning algorithm seeks to find, for each class y_i, a pattern tree that delivers outputs close to 1 for instances $\boldsymbol{x}$ from this class and outputs close to 0 for instances from other classes. This property is indeed a *sufficient* criterion for correct classification, but actually not a *necessary* one. Indeed, according to (2), a prediction is made by combining the outputs of all pattern trees using the arg max operator. Therefore, a prediction is correct as soon as the true class receives the highest score. This does not mean, of course, that the score must be close to 1, while all other scores are close to 0. Trying to comply with this much stronger property will presumably lead to models that are more complex than necessary.

As an illustration, suppose that all classes are correctly characterized by simple linear functions (i.e., the trees have depth 2 and a WA operator as a root node). Combined with arg max, these functions will always produce the correct prediction, even though the outputs will not always be close to 0 and 1, respectively. Instead, more complex, non-linear models will be needed to produce these type of predictions.

To avoid making the learning problem more difficult than necessarily, we shall propose an alternative formalization in the next section.

3 Co-Evolutionary Pattern Tree Induction

3.1 Co-Evolutionary Algorithms

Evolutionary algorithms (EA) are population-based stochastic search methods which seek to optimize a solution by mimicking the process of biological evolution. They can be applied in a quite universal way and have been used in a wide spectrum of application domains.

To apply evolutionary algorithms to complex problems more efficiently, a modularization technique, referred to as *co-evolution*, has recently been proposed [Potter and Jong, 2000; Morrison and Oppacher, 1999]. The general idea of co-evolution is to evolve the sub-components of a (structured) solution, also referred to as *species*, in different sub-populations. To assure that the sub-components can be assembled into a globally optimal solution, the fitness of an individual in a species is evaluated by its ability to participate in a cooperative team consisting of one representative per species. The global fitness function used in this context is also referred to as the *shared domain model.*

Determining the fitness of individuals of a certain species can simply be achieved by the evaluation of collaborations formed with representatives from each of the other species. Representatives of a species can be individuals of a certain fitness, or even the whole population. Essential for the validity of an individuals fitness, which can also be seen as a measure of the individual's contribution to the overall solution, is the selection of representatives of the other species.

3.2 Pattern Tree Induction

In our concrete application of pattern tree induction, an individual is a single pattern tree. We evolve one species per class and denote by $I_i^{(t)}$ the t-th generation of the i-th species. A hypothesis h consists of exactly one individual for each species, that is, one pattern tree for each class. As a fitness criterion (shared domain model), we use the classification accuracy of a hypothesis on the training data.

The evolutionary process comprises the following steps:

1. Initialize each species
2. Evaluate collective fitness
3. Check termination condition
4. Reproduce each species
5. Mutate each species
6. Continue at 2

Step 1 - Initialization

To obtain a first generation of pattern trees, random pattern trees of size 1 or 3 are created. Here, the size of a tree is defined as the number of nodes in the tree, including both, internal and leaf nodes.

Step 2 - Evaluation

After a new generation has been created, the fitness of all individuals of each species must be calculated. This is done by building every possible classifier, that is, every combination

$$h_{j_1,\ldots j_k}^{(t)} \overset{\text{df}}{=} (pt_{1,j_1}^{(t)}, pt_{2,j_2}^{(t)}, \ldots, pt_{k,j_k}^{(t)}) \in I_1^{(t)} \times I_2^{(t)} \times \ldots \times I_k^{(t)}$$

of pattern trees with k denoting the number of species (classes) and pt_{i,j_i} being the j_i-th pattern tree of the i-th species. The number of possible combinations is m^k, with m the size of each population (we evolve each species using the same population size).

As mentioned above, a hypothesis h is evaluated by its accuracy on the training data, acc(h). Moreover, the evaluation (fitness) of a single pattern tree $pt_{j_i}^{(t)}$ is given by the best hypothesis in which it has participated:

$$F\left(pt_{j_i}^{(t)}\right) \overset{\text{df}}{=} \max_{j_1,\ldots,j_{i-1},j_{i+1},\ldots,j_k} \operatorname{acc}\left(h_{j_1,\ldots,j_k}^{(t)}\right) \quad .$$

Step 3 - Termination

The proposed Co-EA terminates if one of the following conditions hold. First, the iteration stops after a maximum number of iterations. Second, the Co-EA also stops if no significant improvement can be achieved during a certain number of iterations.

Therefore, two thresholds δ and d have been introduced to track the accuracy improvement of the most accurate (best) hypothesis h_{best}^t of each generation. If the condition

$$acc(h_{best}^t) + \delta \geq acc(h_{best}^{t+l})$$

holds for all $l \in \{1, \ldots, d\}$, iteration stops.

Step 4 - Reproduction

Reproduction in terms of EAs means the creation of a new generation. Therefore, individuals of the current (parent) generation are selected at random, with a probability proportional to their fitness, to form a set of parents.

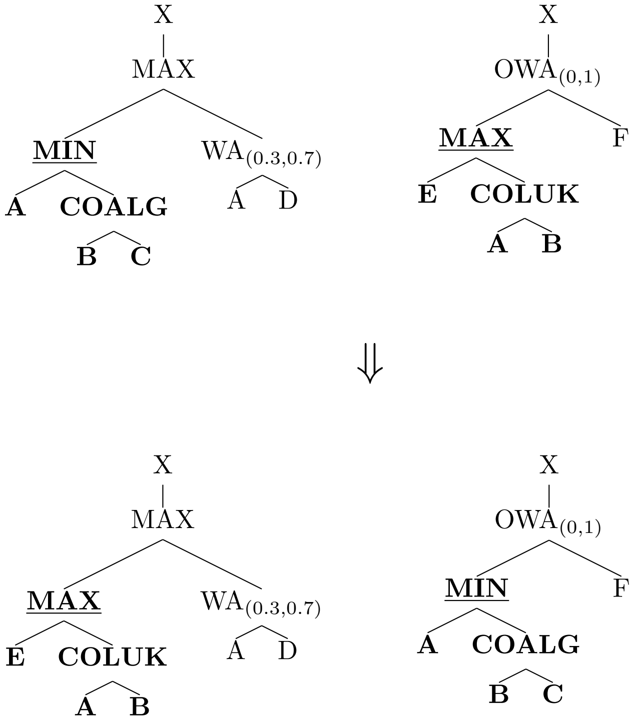


Figure 2: Cross-Over Operator

To make sure that the best solution found so far is not lost, elitist selection is applied, which means that the best hypothesis of each generation is directly transfered to the new generation.

Given two individuals $pt^{(t)}$ and $pt'^{(t)}$, a cross-over operator on trees is used for reproduction, that is, to create their children $pt^{(t+1)}$ and $pt'^{(t+1)}$. To this end, one node within each tree is chosen at random and the corresponding subtrees are interchanged. Fig. 2 illustrates this operation by means of an example.

Step 5 - Mutation
To guarantee a proper level of diversity, a set of randomly chosen individuals are mutated after reproduction. The probability of an individual being chosen for mutation is uniform over the population and determined by the predefined mutation rate. If an individual has been chosen for mutation, one of three different mutation operators is applied. The selection of the mutation operator is again equally random.
Mutate Leaf: Randomly selects a leaf node and replaces it by a randomly chosen other leaf node. This comes down to replacing a fuzzy term F_{ij} of an attribute A_i by another term F_{lj} of another attribute A_l.

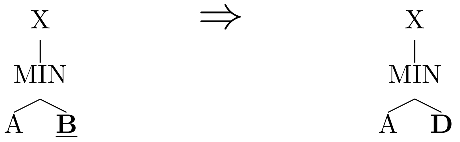


Figure 3: Example Mutate Leaf

Mutate Operator: Randomly selects an internal node and replaces it with a randomly chosen different one. Since an internal node represents a fuzzy operator, this mutation changes the type of aggregation of its children. In the experiments presented in Section 4, we used the t-norms and t-conorms already presented in Table 1 and, moreover, the WA and OWA operators with weight vectors $(w_1, w_2) \in \{(0,1), (0.2, 0.8), ..., (1,0)\}$.

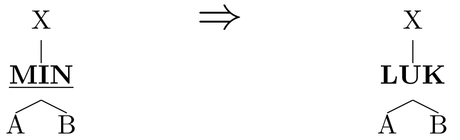


Figure 4: Example Mutate Operator

Mutate Tree: Randomly selects a subtree and replaces it by a new, randomly created tree. This operator somehow combines the first and the second one.

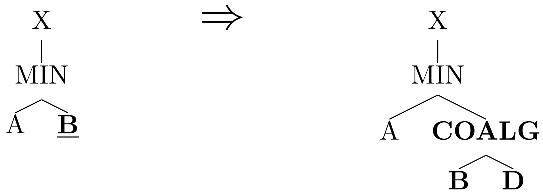


Figure 5: Example Mutate Tree

4 Experiments

We have compared the original pattern tree algorithm (PT) with our co-evolutionary variant (CPT) on a number of benchmark data sets; see Table 2 for a summary of the data sets and their properties (number of classes (#C), number of instances (#I), number of numerical (#num) and nominal attributes (#nom)).

All attributes have been fuzzified in the following way. For nominal attributes, each possible value v was encoded as a single fuzzy set $Term_v$:

$$Term_v(x) = \begin{cases} 1 & x = v \\ 0 & otherwise \end{cases}$$

For a numeric attribute with *min* and *max* being the minimum and the maximum value in the training data, two fuzzy terms *Low* and *High* have been created. The corresponding membership functions are defined as follows:

$$Low(x) = \begin{cases} 1 & x < min \\ 0 & x > max \\ 1 - \frac{x - min}{max - min} & otherwise \end{cases}$$

$$High(x) = \begin{cases} 1 & x > max \\ 0 & x < min \\ \frac{x - min}{max - min} & otherwise \end{cases}$$

These fuzzy sets allow for modeling two types of influence of an attribute on the class membership, namely a positive and a negative one. (Note that all operators appearing at inner nodes of a pattern tree are monotone increasing in their arguments.)

Both PT and CPT were implemented under the WEKA Machine Learning Framework [Witten and Frank, 2005]. We used the following parametrization: Mutation rate 0.3, population size between 5 and 20

Dataset	#C	#I	#num	#nom
Australian	2	690	6	9
Authorship	4	841	69	1
Blood	2	748	4	1
Cancer	2	683	9	1
CMC	3	1473	2	8
Credit	2	690	6	10
German	2	1000	7	14
Haberman	2	306	3	1
HallOfFame	3	1320	15	2
Heart	2	270	7	7
Ionosphere	2	351	34	1
Iris	3	150	4	1
Vehicle	4	846	18	1
Wine	3	178	13	1

Table 2: Data sets and their properties

data set	PT	CPT	C4.5
Australian	85.2174	**85.7971**	86.0870
Authorship	**97.5030**	96.6706	93.6980
Blood	77.0053	**78.8770**	77.8075
Cancer	96.1933	**97.0717**	96.0469
CMC	52.4779	**53.4284**	52.1385
Credit	85.2174	**85.9420**	85.9420
German	72.4000	**72.6000**	70.7000
Haberman	73.2026	**75.4902**	71.8954
HallOfFame	92.3485	**92.5758**	92.8788
Heart	81.4815	**81.8519**	80.0000
Ionosphere	89.4587	**90.8832**	91.4530
Iris	**96.6667**	94.0000	96.0000
Vehicle	61.1111	**61.9385**	72.5768
Wine	**96.0674**	93.8202	93.8202

Table 3: Average accuracy of PT and CPT in a 10-fold cross validation study. The best result among these two in marked in bold font. Additionally, results are shown for C4.5.

(depending on the number of classes), accuracy improvement thresholds $\delta = 0.001$ and $d = 1000$, maximum number of overall iterations 5000.

Table 3 shows the results of a 10-fold cross validation. As a non-competitor, we also included the WEKA implementation of the well-known C4.5 classifier [Quinlan, 1993], just to convey an idea about the absolute performance of pattern tree induction in comparison to state-of-the-art methods. As can be seen from the results, in a direct comparison, CPT is superior and outperforms PT most of the time. In terms of a simple sign test applied to the win/loss statistic, the superiority of CPT is indeed significant at a 10% level. Yet, the differences in classification accuracy are often rather small, and on average, they are actually smaller than expected. Thus, in summary, the experiments show that there is indeed scope for improving the simple greedy strategy underlying PT. On the other hand, they also show that, in light of its purely heuristic nature, this strategy performs comparatively well. Moreover, one has to consider that the increase in predictive accuracy achieved by CPT comes at the price of a significantly higher runtime. In fact, it is well known that evolutionary optimization is rather expensive from a computational point of view.

5 Conclusions and Future Work

In this paper, we have developed an alternative method for learning pattern tree classifiers. This work was mainly motivated by two alleged disadvantages of the original learning method: First, it is based on a simple greedy and, therefore, myopic search strategy, which is likely to get stuck in local optima. Second, it seems to solve a problem which is actually more difficult than necessary.

To overcome these disadvantages, we have employed a co-evolutionary algorithm as a more sophisticated search method. Moreover, instead of maximizing a similarity function for each class separately, we take the interdependency of the individual pattern trees into account and seek to maximize classification accuracy directly.

The experimental results are in a sense ambivalent. On the one hand, they show that our approach is indeed able to improve predictive accuracy, albeit at the cost of an increased runtime. Thus, it confirms our presumption that the original learning algorithm is not optimal. On the other hand, the gains are not as high as we expected. This, of course, can have different reasons. First, one cannot exclude that, despite its obvious shortcomings, the original learning method does produce good models. In this case, it would be interesting to find out why it actually works. Moreover, it is of course possible that our approach is still far from optimal, and that better methods can be developed.

These issues will be addressed in future work. Besides, we plan to apply pattern trees to learning problems beyond simple classification, such as label ranking [Hüllermeier *et al.*, 2008], for which they seem to be especially appealing. Finally, we are interested in analyzing the robustness of pattern tree induction, that is, the question of how sensitive the topology and parametrization of the learned trees is toward variations of the data. This question is especially important from an interpretation point of view.

References

[Huang *et al.*, 2008] Z. Huang, T.D. Gedeon, and M. Nikravesh. Pattern tree induction: A new machine learning method. *IEEE Transactions on Fuzzy Systems*, 16(4):958–970, 2008.

[Hüllermeier *et al.*, 2008] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172:1897–1917, 2008.

[Klement *et al.*, 2002] EP. Klement, R. Mesiar, and E. Pap. *Triangular Norms*. Kluwer Academic Publishers, 2002.

[Morrison and Oppacher, 1999] Jason Morrison and Franz Oppacher. A general model of co-evolution for genetic algorithms. In *In Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA '99), (Portoroz*, pages 262–268. Springer-Verlag, 1999.

[Potter and Jong, 2000] M.A. Potter and K.A. De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8:1–29, 2000.

[Quinlan, 1993] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[Schweizer and Sklar, 1983] B. Schweizer and A. Sklar. *Probabilistic Metric Spaces*. North-Holland, New York, 1983.

[Witten and Frank, 2005] IH. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2 edition, 2005.

[Yager, 1988] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. 18(1):183–190, 1988.

Analyzing the regulation of metabolic pathways in human breast cancer

Eva Maria Surmann[1,+], Gunnar Schramm[1,+], Stefan Wiesberg[2], Marcus Oswald[2], Gerhard Reinelt[2], Roland Eils[1,3,*] & Rainer König[1,*]

[1] Department of Bioinformatics and Functional Genomics, Institute of Pharmacy and Molecular Biotechnology, and Bioquant, University of Heidelberg, INF 267, 69120 Heidelberg, Germany
[2] Interdisciplinary Center for Scientific Computing, University of Heidelberg, 69120 Heidelberg
[3] Theoretical Bioinformatics, German Cancer Research Center, INF 580, 69121 Heidelberg
r.koenig@dkfz.de, r.eils@dkfz.de
[+] These authors have contributed equally
[*] Corresponding author

Abstract

Tumor therapy mainly attacks the metabolism to interfere their anabolism and signaling of proliferative second messengers. However, the metabolic demands of different cancers are very heterogenous and depend on their origin of tissue, age, gender and other clinical parameters. We wanted to find out tumor specific regulation for the metabolism of breast cancer. For this, we mapped gene expression data from microarrays onto the corresponding enzymes and their metabolic reaction network. We used Haar Wavelet transforms on optimally arranged grid representations of metabolic pathways as a pattern recognition method to detect orchestrated regulation of neighboring enzymes in the network. Significant combined expression pattern were used to select metabolic pathways showing shifted regulation of the aggressive tumors. Besides up-regulation for energy production and nucleotide anabolism, we found an interesting cellular switch in the interplay of biosynthesis of steroids and bile acids. The biosynthesis of steroids was up-regulated for estrogen synthesis which is needed in breast cancer for proliferative signaling. In turn, the decomposition of steroid precursors was blocked by down-regulation of the bile acid pathway. In conclusion, we applied an intelligent pattern recognition method on networks and elucidated substantial regulation of human breast cancer pointing to specific treatment.

1 Introduction

Breast cancer is a prevalent disease and a leading cause of cancer death in women [Oakman and Di Leo, 2009]. Worldwide, breast cancer is the second most common type of cancer after lung cancer and the fifth most common cause of cancer death. Breast cancer patients with the same stage of disease can have very different treatment responses and overall outcome. Clinical predictive factors like age, tumor size, lymph node status, histological and pathological grade or hormone-receptor status, often fail to accurately predict clinical outcome, distant metastasis and recurrence of the cancer. Chemotherapy and hormonal therapy reduces the risk of distant metastases by approximately one third. However, 70-80% of the patients would have survived without it [Van 'T Veer, *et al.*, 2002]. A more accurate means of prognosis and selection of therapy would substantially improve disease-free and overall survival of breast cancer patients [Van De Vijver and Bernards, 2002]. Cancer cells acquire their hallmarks of malignancy through the accumulation of advantageous gene activation and inactivation events over long periods of time [Fan and Perou, 2006]. Nevertheless, the molecular basis of breast cancer tumorigenesis remains to be poorly understood. A long-standing strategy for cancer treatment is to attack basic tumor metabolism by inhibiting nucleotide biosynthesis [Chen, 2007, Lui, 1982] and DNA production [Pedersen-Bjergaard, 1985]. Besides this, over the past decade there have been exciting developments in analyzing large scale gene expression profiles. It improved the understanding of the tumors' composition and behavior to develop new targets for therapy [Oakman and Di Leo, 2009]. Many studies of gene expression have identified expression profiles that are prognostic for patients with breast cancer. However, comparisons of the lists of genes derived from these studies showed that they overlap only slightly due to differences in the patient cohorts, microarray platforms, and mathematical methods of analysis [Fan and Perou, 2006]. One strategy to tackle this problem is to map lists of differentially expressed genes on groups of genes with related functions according to the information provided by several databases such as Gene Ontology [Harris, *et al.*, 2004] and KEGG [Kanehisa, *et al.*, 2008]. Finding enrichments of specific gene sets related to certain phenotypes or cell states yields a functional grouping of differentially expressed genes which can be related to their pathogenic behavior and can lead to more robust results in comparison to the analysis of single genes. To detect the enrichment of gene sets, commonly, a list of significantly differentially expressed genes is identified and statistical tests applied, such as Fisher's exact test and χ^2 test. In a different approach, a gene-specific statistics, known as the "local" statistics, measures the strength of association between the gene expression and the phenotype for each gene. A global statistics for a gene set is then constructed as a function of local statistics for each gene in it. The significance is assessed by permutation tests [Pitman, 1937]. A global test for gene sets to associate gene expression with clinical outcome was presented by Goeman and co-workers and enabled determining whether the global expression pattern of a group of genes is significantly related to clinical outcome of interest using a linear regression approach [Goeman, *et al.*, 2004]. In another approach, expression levels of all genes in the gene sets are combined and presented as gene specific features. These features are then compared between the treatment and the control groups to identify significantly affected gene sets [Yan and Sun, 2008]. In general, these methods test the association of all genes in a gene set with the phenotypes, whereas often only genes in a subset of the gene set are associated with the phenotype. Some of the genes may not belong to the set due to incompleteness or erroneous in the available data. Additionally, even if all genes in the gene set have apparently the same function, or belong to the same process, it is likely that only a few genes are associated with the phenotype. To overcome these and other gene-associated problems, transcriptional data was analyzed using topology information of cellular networks. Topological information derived from the metabolic network was connected by calculating Z-scores of highly correlated sub-networks [Patil and Nielsen, 2005]. Chuang and co-workers improved classification of breast cancers with expression patterns of small subnets of a signal transduction network [Chuang, *et al.*, 2007]. Substantial new genetic mediators for prostate cancer were found using reverse engineered gene networks in combination with gene expression profiles [Ergun, *et al.*, 2007]. Common gene expression levels of neighboring nodes in a metabolic network were calculated by averaging over all neighbors of a gene, and revealed several interesting regulated pathways for the human immune system [Nacu, *et al.*, 2007]. However, these approaches were not developed to detect highly contrasting expression of neighboring genes that undergo a switch-like shift of regulation in a tumor cell. Importantly, especially these switches can be highly relevant to identify potential drug targets that specifically attack the tumor at nodes at which it rewires the network to establish parasitic advantages.

In our study, gene expression profiles of breast tumors having an “unfavorable” prognosis were compared to breast tumors with a “favorable” prognosis. We wanted to track how the aggressive (unfavorable) tumors have specifically regulated their metabolism to optimize their oncogenetic fitness, and to elucidate ways to severely perturb this process. For this, we used an approach that finds orchestrated regulation of neighboring enzymes in the metabolic network. We mapped the gene expression data onto optimally arranged grid representations of pathways of the metabolic network and applied Haar wavelet transforms onto defined pathways of the network to combine gene expression values from neighboring enzymes. These combined features were tested with a rank product test (Wilcoxon) if they could separate samples from different treatments. Metabolic pathways were selected that had features with the most discriminative gene expression patterns. We detected a substantially higher number of significant gene expression patterns in comparison to commonly used enrichment tests. We revealed 14 significant metabolic pathways including increased purine and pyrimidine biosynthesis which were needed for increased mitosis cycles. Furthermore, we found pathways for increased energy metabolism (glycolysis, pyruvate metabolism and fructose/mannose metabolism). Interestingly, the observed regulation revealed a cellular switch in the pathway for cholesterol synthesis to direct the metabolic flux for synthesis of steroids while preventing degradation into bile acids.

2 Results and Discussion

In this study, 275 patients were examined, 196 having a “favorable” and 79 patients an “unfavorable” prognosis. 1826 reactions could be extracted from KEGG [Kanehisa, *et al.*, 2008] for 1771 out of which expression values could be assigned. The workflow of the method is depicted in Figure 1. Pathway maps from KEGG were represented as two-dimensional lattice grids with dense packed reactions. The reactions were arranged in a way that their neighborhoods in the network were preserved as optimal as possible (using the grid arrangement method, for details see methods). Gene expression data was mapped onto the reactions representing the according expressed enzymes.

Wavelet transforms were used to combine expression values of neighboring reactions by all possible combinations of substractions and additions. The out coming features were tested (Wilcoxon rank test) for their possibility to discriminate between the two tumor entities (favorable and unfavorable). Figure 2 illustrates the principle for a schematic pathway. Pathways with the best discriminating features were selected.

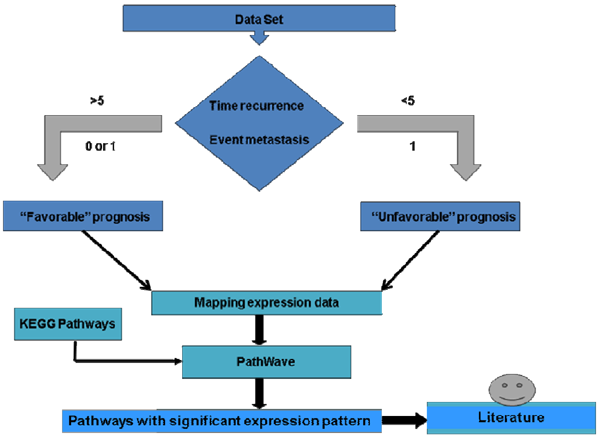


Figure 1. The workflow. Samples were divided into favorable and unfavorable prognosis according to their time recurrence (time period of relapse after the first event of breast cancer, denoted in years) and event of metastasis (= 0 if no metastasis within the first 5 years, =1 else). Expression data were mapped onto the reactions from KEGG and analyzed. Pathways with significant expression pattern were ranked according to the significance of the pattern and compared with the literature.

We revealed significant features from 14 different pathways (Table 1), including pyrimidine, purine, amninoacyl-tRNA metabolism, pyruvate metabolism, glycolysis/gluconeogenesis and fructose/mannose metabolism (red in Table 1). The pathways have been expected as they accounted for higher biosynthesis of nucleic acids and proteins and higher energy demands of the aggressive (unfavorable) tumors. We also revealed less expected differentially regulated pathways, such as biosynthesis of steroids and bile acids.

We compared the performance of our algorithm with commonly used enrichment methods. Fisher’s exact tests revealed only one pathway to be significantly enriched with differentially regulated genes ($P \leq 0.05$, threshold for defining the differentially expressed genes also P=0.05). Fisher’s exact tests yielded the pyrimidine pathway to be significant (P=8.69E-3). In addition, we applied the well established Gene Set Enrichment Analysis (GSEA, two-sided, see [Mootha, *et al.*, 2003]) to the data yielding two significantly enriched pathways, again the pyrimidine metabolism (P < 1E-17) and additionally the biosynthesis of unsaturated fatty acids (P = 0.0297). Note, that both pathways showed also up with our method. In the following, we will discuss the oncogenetic relevance of the pathways we found.

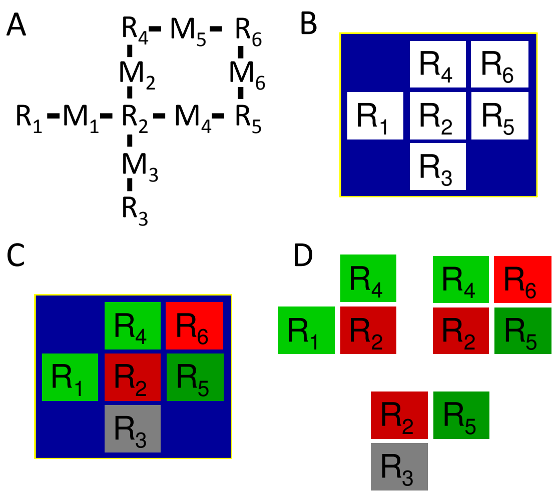


Figure 2. Extracting the features (schematic view). A. A bipartite graph consisting of metabolites (M) and reactions (R) was assembled using the pathway information from KEGG. B. Reactions were optimally arranged with the grid arrangement method which optimally preserves next nearest neighborhoods while minimizing the size of the grid. C. Gene expression data was mapped onto the corresponding enzymatic reactions. D. Combined gene expression features were assembled by Haar wavelet transforms which basically (in the 1st level) calculated additive and substractive combinations of 2x2 pixels of the grid (pixels without reactions were filled with zeros). The same procedure was done for all tumor samples. The feature which best separated the tumor entities (favorable from unfavorable) was selected for the significance of this pathway.

2.1 Pyrimidine and purine metabolism

Most up-regulated reactions were identified in the pyrimidine (P=9.47E-5) and purine (P=1.41E-3) metabolism. These pathways were up-regulated to enable enforced nucleotide biosynthesis for increased cell cycle activity of the aggressive tumors. Nearly all enzymes involved in the biosynthetic pathway for nucleotides were up-regulated, such as enzymes converting substrates to dNTPs and polyribonucleotide nucleotidyltransferases (EC 2.7.7.7), incorporating dNTPs into DNA. Enzymes reversing pyrimidine and purine anabolism, such as enzymes degrading dNTPs, were down-regulated (EC 3.6.1.17, 1.3.1.2, 2.7.4.3 and 6.3.5.3). Reactions involved in RNA synthesis were partially up-regulated to increase protein biosynthesis (EC 2.7.7.8). Furthermore, reactions which were responsible for synthesizing adenosine were up-regulated (EC 2.4.2.1). Adenosine was shown to be angiogenic, cyto-protective and anti-inflammatory in several tissues and contributed to more aggressive behavior and metastasis of cancer cells [Spychala, 2003].

2.2 Pathways for energy supply were significantly up-regulated

We detected significant differential expression patterns in glycolysis (P=2.94E-3), pyruvate (P=4.13E-3) and fructose/mannose (P=2.24E-2) metabolism. They were mostly up-regulated to generate sufficient energy for fast-growing cancer cells under oxygen limitation. Genes of the glycolysis pathway have been found to be overexpressed in a set of 24 cancers, which is special because other pathways showed less consistent up-regulation so far [Altenberg and Greulich, 2004]. Glycolysis is increased in cancers to generate ATP, known as the Warburg effect as a consequence of hypoxia in the tumor environment. Under normal conditions, ATP is generated through oxidative phosphorylation which is more efficient compared to glycolysis. Is this efficient way compromised, alternative metabolic pathways such as increasing glycolytic activity are adapted to maintain energy supply [Xu and Huang, 2005]. Actually, the amount of up-regulated genes is distinct in different cancer types. In our study all except one differentially regulated reactions were up-regulated, including enzymes such as glyceraldehyde-3-phosphate dehydrogenase (EC 1.2.1.12), phosphor glycerate mutase (EC 5.4.2.1), phosphoglycerate kinase (EC 2.7.2.3), triosephosphate isomerase (EC 5.3.1.1) and fructose-bisphosphate aldolase

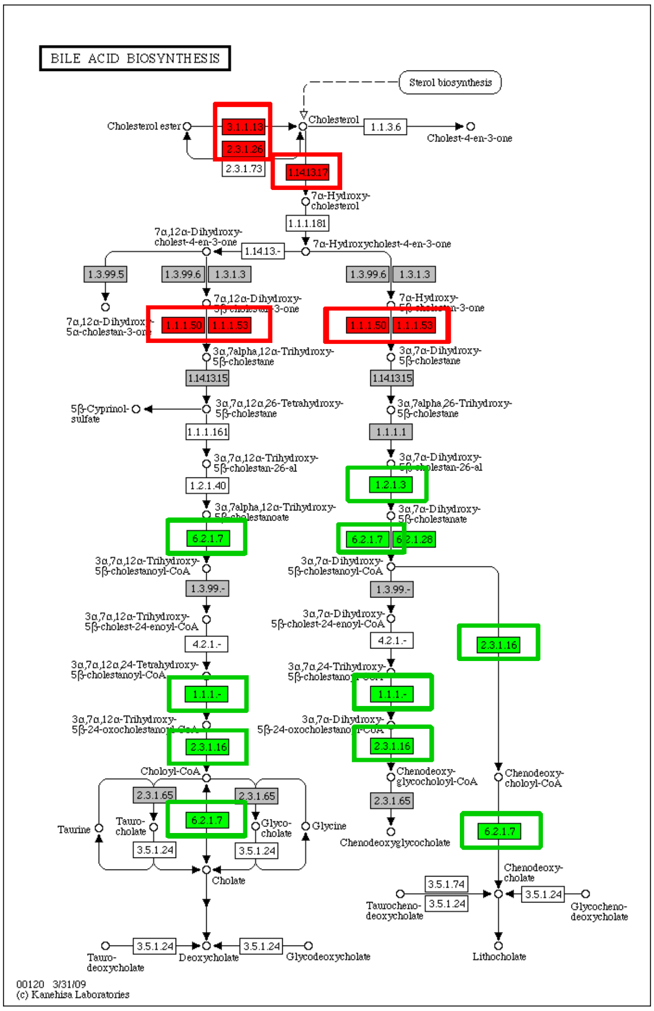
Table 1. Identified significant differentially regulated pathways with more than three differentially regulated KEGG reactions; p-value threshold = 0.05.

Ranking	Pathway	Differentially regulated	all	Downregulated in bad	Upregulated in bad	pval
1	**Fatty acid metabolism**	14	36	9	5	4.03E-005
2	**Pyrimidine metabolism**	39	75	3	36	9.47E-005
3	**Alanine and aspartate metabolism**	9	19	1	8	3.74E-004
4	**Biosynthesis of steroids**	7	41	1	6	4.66E-004
5	**Aminoacyl-tRNA biosynthesis**	5	20	1	4	5.47E-004
6	**Bile acid biosynthesis**	16	28	10	6	5.68E-004
7	**Valine, leucine and isoleucine degradation**	6	36	5	1	8.35E-004
8	**Purine metabolism**	40	91	6	34	1.41E-003
9	**Tryptophan metabolism**	5	39	2	3	2.13E-003
10	**Glycolysis / Gluconeogenesis**	6	32	0	6	2.94E-003
11	**Biosynthesis of unsaturated fatty acids**	19	41	2	17	3.16E-003
12	**Pyruvate metabolism**	6	27	2	4	4.13E-003
13	**Inositol phosphate metabolism**	7	15	0	7	4.75E-003
14	**Fructose and mannose metabolism**	11	23	2	9	2.24E-002

(EC 4.1.2.13). These enzymes were also described as up-regulated by Altenberg and co-workers [Altenberg and Greulich, 2004]. Inhibitors of glycolysis, such as the glucose analogue 2-deoxyglucose, which binds and suppresses hexokinase I, and arsenate, that causes arsenolysis in glyceraldehyde-3-phosphate dehydrogenase, as well as 3-bromopyruvate, an inhibitor of hexokinase II, have already been developed to target this metabolic abnormality and could effectively kill cancer cells [Xu and Huang, 2005]. Inhibition of glycolysis was also effective in killing cancer cells with a multidrug resistance (MDR) phenotype. It is known that cells expressing MDR proteins require ATP as their energy source to export the drugs out of the cell. Thus, to overcome this drug resistance, depletion of cellular ATP causes the excretion to fail and consequently, cancer cells become more sensitive to anti-cancer therapy [Xu and Huang, 2005]. We found a smaller pattern of up-regulated reactions in the pyruvate metabolism. Significantly up-regulated were: pyruvate kinase (EC 2.7.1.49) which is responsible for ATP production in glycolysis, acetyl-CoA hydrolase (EC 3.1.2.1) which hydrolyses acetyl-CoA to acetate and CoA, and pyruvate carboxylase (EC 6.4.1.1) for production of oxaloacetate out of pyruvate. Aldehyde dehydrogenase (EC 1.2.1.3) was down regulated to prevent degradation of fatty acids. Oxaloacetate was reported to directly induce cell proliferation by increasing DNA synthesis [Li, 1993]. Surprisingly, our results did not show any increase in lactate dehydrogenase, which is often up-regulated in cancer cells [Altenberg and Greulich, 2004]. Highly expressed reactions in fructose and mannose metabolism were mainly involved in the conversion of D-fructose (EC 4.1.2.13 and EC 5.3.1.1), D-fructose-1P (EC 2.7.1.1, 2.7.1.11, 4.1.2.13) and D-mannose (EC 2.7.1.1, 2.7.1.11, 4.1.2.13) into glyceraldehyd-3P, a substrate of glycolysis, leading to increased ATP production.

2.3 Biosynthesis of steroids and bile acids

The steroid pathway was mainly up-regulated in breast cancer of unfavorable outcome. 9 out of 10 reactions were significantly up-regulated (EC 2.7.1.36, 4.1.1.33, 2.5.1.29, 2.4.1.10, 2.5.1.1, 1.14.99.7, 1.1.1.70, 5.3.3.5, 1.3.1.21). Cholesterol is a precursor for the biosynthesis of steroid hormones. It was shown that sex steroid hormones such as estrogen increase the proliferation of breast cancer cells by acting on estrogen receptors [Suzuki and Sasano, 2005]. Directly connected with the production of cholesterol is the biosynthesis of bile acids (Figure 3). In the bile acid biosynthesis pathway, reactions generating cholesterol and its derivates were up-regulated (EC 3.1.1.13 and 2.3.1.26), whereas the lower part of the pathway was mainly down-regulated, i.e. the production of bile acids such as cholate and lithocholate (EC 6.2.1.7, 1.1.1.-, 2.3.1.16, 6.2.1.28). Conversion of cholesterol to bile acids is the major pathway for cholesterol catabolism in the human body [Zimber and Gespach, 2008]. Therefore, down-regulation of the lower part of the bile acid pathway and up-regulation of cholesterol biosynthesis may, in conjunction, support steroid biosynthesis which supports estrogen mediated tumorigensis of the breast cancer cells. Besides this, there have been intensive cancer investigations on bile acids as they have very heterogenous effects on tumor cells and carcinogenesis. They are known enhancers of invasiveness of colon cancers [Debruyne, *et al.*, 2001, Pai, *et al.*, 2004].


BILE ACID BIOSYNTHESIS
Sterol biosynthesis
Cholesterol ester
3.1.1.13
2.3.1.26
2.3.1.73
Cholesterol
1.1.3.6
Cholest-4-en-3-one
1.14.13.17
7α-Hydroxy-cholesterol
1.1.1.181
7α,12α-Dihydroxy-cholest-4-en-3-one
1.14.13.-
7α-Hydroxycholest-4-en-3-one
1.3.99.5
1.3.99.6
1.3.1.3
1.3.99.6
1.3.1.3
7α,12α-Dihydroxy-5β-cholestan-3-one
7α-Hydroxy-5β-cholestan-3-one
7α,12α-Dihydroxy-5α-cholestan-3-one
1.1.1.50
1.1.1.53
1.1.1.50
1.1.1.53
3α,7alpha,12α-Trihydroxy-5β-cholestane
3α,7α-Dihydroxy-5β-cholestane
1.14.13.15
1.14.13.15
5β-Cyprinol-sulfate
3α,7α,12α,26-Tetrahydroxy-5β-cholestane
3α,7alpha,26-Trihydroxy-5β-cholestane
1.1.1.161
1.1.1.1
3α,7α,12α-Trihydroxy-5β-cholestan-26-al
3α,7α-Dihydroxy-5β-cholestan-26-al
1.2.1.40
1.2.1.3
3α,7alpha,12α-Trihydroxy-5β-cholestanoate
3α,7α-Dihydroxy-5β-cholestanate
6.2.1.7
6.2.1.7
6.2.1.28
3α,7α,12α-Trihydroxy-5β-cholestanoyl-CoA
3α,7α-Dihydroxy-5β-cholestanoyl-CoA
1.3.99.-
1.3.99.-
3α,7α,12α-Trihydroxy-5β-cholest-24-enoyl-CoA
3α,7α-Dihydroxy-5β-cholest-24-enoyl-CoA
4.2.1.-
4.2.1.-
2.3.1.16
3α,7α,12α,24-Tetrahydroxy-5β-cholestanoyl-CoA
3α,7α,24-Trihydroxy-5β-cholestanoyl-CoA
1.1.1.-
1.1.1.-
3α,7α,12α-Trihydroxy-5β-24-oxocholestanoyl-CoA
3α,7α-Dihydroxy-5β-24-oxocholestanoyl-CoA
2.3.1.16
2.3.1.16
Choloyl-CoA
Chenodeoxy-glycocholoyl-CoA
Chenodeoxy-choloyl-CoA
2.3.1.65
2.3.1.65
2.3.1.65
Taurine
Tauro-cholate
6.2.1.7
Glyco-cholate
Glycine
Chenodeoxyglycocholate
6.2.1.7
3.5.1.24
3.5.1.24
Cholate
Chenodeoxy-cholate
3.5.1.74
3.5.1.24
3.5.1.24
Taurocheno-deoxycholate
Glycocheno-deoxycholate
Tauro-deoxycholate
3.5.1.24
Deoxycholate
3.5.1.24
Glycodeoxycholate
Lithocholate

00120 3/31/09
(c) Kanehisa Laboratories

Figure 3. Regulation of the pathway for bile acid biosynthesis. Red indicates up-regulation, blue down-regulation, and grey no differential regulation. Framed reactions were detected with our method as a significant differential expression pattern. The map was taken from KEGG [Kanehisa, *et al.*, 2008]. The lower part of this pathway was down-regulated to prevent degradation of steroids into bile acids, whereas the upper part was up-regulated to support steroid metabolism.

In turn, bile acids can induce apoptosis either specifically (receptor-mediated interactions) [Garewal, *et al.*, 1996] or, in high concentrations, non-specifically (as detergents) through mitochondrial destabilization and oxidative stress [Katona and Stenson, 2009]. Non-toxic doses of deoxycholic acid (DCA), chenodeoxycholic acid (CDCA) and lithocholic acid (LCA) induced differentiaion in promyelocytic leukemia cell lines [Zimber, *et al.*, 1994]. 6ECDCA is a synthetic bile acid derivative and can act as a selective FXR ligand to promote differentiaion of preadipocyte cell lines [Zimber, *et al.*, 1994]. In the blood plasma of postmesopausal women with newly diagnosed breast cancer, elevated concentration of DCA was detected which may have been released from osteoblasts to induce migration of the cancer (see [Zimber and Gespach, 2008]). This would explain a down-regulation in breast cancer cells as the may use elevated bile acid levels from the blood for detaching and infiltrating while sustaining their endogenous cholesterol synthesis. Bile acids are normally predominantly produced by hepatocytes which may explain the elevated levels of bile acids in the blood when elevated breast cancer estrogen is decomposed into bile acids in the liver. In conclusion, this regulation may have revealed a cellular switch to direct the metabolic flux for the conversion of cholesterol into steroid hormones which is more beneficial for the tumor than the production of bile acids, suggesting drug targets in the biosynthesis of steroids, such as mevalonate (diphospho) decarboxylase (EC 4.1.1.33) and mevalonate kinase (EC 2.7.1.36).

3 Conclusions

Performing a network specific expression analysis revealed interesting insights into the metabolism and regulation of aggressive breast tumor cells. Expected differentially regulated pathways in cancer could be confirmed, e.g. the aggressive tumors showed significantly up-regulated pathways for purine and pyrimidine synthesis to maintain elevated proliferation, as well as the up-regulation of glycolysis and pyruvate metabolism to supply energy for the tumor. The analysis revealed insights into differentially regulated metabolic pathways in breast cancer cells, which were responsible for the induction of proliferation by inositol signal transduction cascades (inositol phosphate metabolism) and steroid hormones (biosynthesis of steroids). An interesting view in breast cancer metabolism was offered by the interplay between biosynthesis of steroids and bile acids, the latter of which was down-regulated in order to convert cholesterol into proliferative acting steroid hormones. Such a cellular switch would be interesting to compare to other cancer and tissues, also in respect to define a specific therapy. Performing Fisher's exact tests as a standard enrichment analysis revealed the pyrimidine biosynthesis pathway to be significantly enriched with differentially expressed genes. The GSEA method yielded pyrimidine biosynthesis and biosynthesis of unsaturated fatty acids. In addition to these pathways, our method yielded twelve further highly significant regulation patterns, showing its increased sensitivity in relation to standard enrichment tests. However, some pathways in our results, such as valine, leucine and isoleucine degradation, could not be associated with oncogenesis and may need further examination. Complex regulated pathways which were relevant to breast tumors with unfavorable prognosis were detected and described in a straightforward manner. The global analysis of network pattern offered a good insight into the regulation of metabolism in breast tumors and may support revealing new potential targets for drug design, especially in the interplay of the biosynthesis of bile acids and steroids. The method was developed to discover metabolic subgraphs or pathways with discriminative gene expression patterns of two sample entities. It can be applied to any two different but comparable expression data sets, like e.g. samples from tumor and normal tissue or two different tumors, cell lines having undergone different treatments, and cell extracts from different time points after treatment. The approach is very general and its applications can be extended to any data of two different entities which can be mapped to a relevant network, as e.g. the www as the network and the usage of its nodes at different time points, respectively. Another example may be analyzing the railway network and people using the train stations at different time points, etc. The method takes advantage from direct neighbor relationships and relationships of local network topology. It may be applied whenever these features are thought to play a major role in the discovery of hot spots in the differences of two different sample populations.

4 Methods

4.1 Preparing the microarray data

Normalized gene expression data was taken from a published study [Van De Vijver and Bernards, 2002] of breast-cancer samples of 295 women (diagnosis between 1984 and 1995) with age ≤ 53 years and no previous history of cancer, except of non-melanoma skin cancer. The gene expression profiles were derived by using oligonucleotide microarrays from Agilent Technologies (www.agilent.de). Data on relapse-free survival (defined as the time to the first event) and overall survival were available for all patients. Most of the patients had breast cancer of stages one and two. 165 had received local therapy alone, 20 had received tamoxifen only, 20 had received tamoxifen plus chemotherapy, and 90 had received chemotherapy only. 151 patients had lymph-node-negative disease and 144 had lymph-node-positive disease. The tumors were primary invasive breast carcinoma that were less than 5 cm in diameter at pathological examination [Van De Vijver and Bernards, 2002]. To differentiate between tumors with "favorable" and "unfavorable" prognosis, samples were separated according to their "time recurrence" (period of time of relapse after the first event of breast cancer, denoted in years) and "event of metastasis" (if metastasis occurs during this period). A time recurrence above 5 years without the event of metastasis indicated a "favorable" prognosis, whereas samples were classified as samples with unfavorable prognosis if they showed a time recurrence of less than five years and the occurrence of metastasis. Samples with ambigious information were discarded and identified by a time recurrence of less than 5 years without the event of metastasis.

4.2 Assembling the metabolic pathways

Pathways were defined according to curated pathway maps of the KEGG database (version from February 4th, 2009) [Kanehisa*, et al.*, 2008]. Each metabolic pathway was established by defining neighbors of reactions as given by the corresponding xml files from KEGG (ftp://ftp.genome.jp/pub/kegg/xml/organisms/hsa). Two reactions were neighbors if a metabolite existed that was the product of one and the substrate of the other. We defined reactions as the nodes and metabolites as the edges between them. Pathways without any connected reaction were discarded. This resulted in 99 pathways with 1826 different reactions. Each pathway was represented by its adjacency-matrix. An entry at row *a* and column *b* was set to one if there existed a metabolite that was produced by reaction *a* and consumed by reaction *b* or vice versa. The sizes of the symmetric adjacency-matrices were between 2x2 and 92x92 reactions.

4.3 Ordering the two-dimensional pathway representation with the grid arrangement method

To apply our feature extraction method we required a 2-dimensional grid arrangement of the metabolic network. We calculated an embedding of the metabolic networks for every KEGG pathway into a 2-dimensional, regular square lattice grid. To preserve neighborhood characteristics of the reactions, we were looking for embeddings in which adjacent nodes of the network were placed onto the grid as close to each other as possible. We wanted to determine an optimal neighborhood in which the total edge length of the graph was minimized according to some metric on the lattice and the network topology was preserved as good as possible. For this purpose the minimization of total edge length is more suitable than a minimization of the longest edge which is widely applied in very large scale integration (VLSI) designs as the latter one allows a variety of optimal solutions in which adjacent nodes are placed unnecessarily far from each other. As a measure of distance in the lattice, we used the natural metric induced by the underlying lattice graph, the so-called Manhattan distance. That is, for any two grid points $u = (i_1, j_1)$ and $v = (i_2, j_2)$ the distance was given by $d_{uv} = | i_1 - i_2 | + | j_1 - j_2 |$. This resulted in an NP-hard combinatorial optimization problem. We stated this problem as an integral linear program (IP) (see [Nemhauser and Wolsey, 1999] for an introduction to integer programming). We formulated the IP by introducing 3-dimensional binary variables x_{vij} for every node *v* and every grid point (*i*, *j*) stating whether or not node *v* has to be placed on grid point (*i*, *j*). For each pair of nodes (*u*,*v*) we calculated their distance d_{uv}. For a given lattice grid *g*, the undirected network graph *G* = (*V*, *E*) with node set *V*, edge set *E* and adjacency matrix *M*, the most basic IP was given by finding an optimum for

$$\min_{x,d} \sum\nolimits_{a,b \in V, a<b} M(a,b) \cdot d_{ab} \ , \tag{1}$$

with the constraints

$$\sum\nolimits_{(i,j) \in g} x_{vij} = 1 \quad \text{for all } v \in V \, , \tag{2}$$

$$\sum\nolimits_{v \in V} x_{vij} \leq 1 \quad \text{for all } (i,j) \in g \tag{3}$$

$$d_{ab} \geq A + B \, , \ d_{ab} \geq A - B \tag{4}$$

$$d_{ab} \geq -A + B \, , \ d_{ab} \geq -A - B \tag{5}$$

for all $(a,b) \in V \times V, a < b$, where

$$A := \sum\nolimits_{(i,j) \in g} i \cdot x_{aij} - \sum\nolimits_{(i,j) \in g} i \cdot x_{bij} \ , \tag{6}$$

$$B := \sum\nolimits_{(i,j) \in g} j \cdot x_{aij} - \sum\nolimits_{(i,j) \in g} j \cdot x_{bij} \ , \tag{7}$$

$$x_{vij} \geq 0, \quad x_{vij} \in \mathrm{Z} \quad \text{for all } v \in V \, , \, (i,j) \in g \tag{8}$$

Constraints (2, 3) guaranteed that all nodes were placed exactly once and that each grid point could be used at most once. Constraints (4, 5) ensured that the distance of node a and b is given by |A| + |B| where A and B are computed by equations (6, 7) as A = i_a - i_b and B = j_a – j_b. All variables were enforced to values 0 or 1 by constraint (8). The problem was solved by CPLEX 8.1 (ILOG, Gentilly, France) for 99 lattice grids (representing 99 KEGG-maps) with an average optimality of 96% for embeddings on square grids of side length $\sqrt{|V|} + 1$, rounded up to the next integer. By choosing a grid of the smallest possible size, we reduced both the number of variables in the model and the number of unoccupied sites on the grid. This basic model was enhanced by a number of graph dependent, additional constraints on the distance variables. They provided lower bounds for the distance sums of well-known subgraph motifs. For an edge induced subgraph $G' \subset G$ with a least objective function contribution of $lb(G')$, the following inequality was added or dynamically separated by

$$\sum_{(u,v) \in E(G')} d_{uv} \geq lb(G'). \tag{9}$$

The right-hand sides $lb(G')$ for the different subgraph motifs needed to be determined only once as they are independent of G. Furthermore, the motifs were defined in a pre-processing step and could therefore be separated quickly during the optimization process. We considered the sub-graph motifs of star graphs, cliques consisting of up to 10 vertices and odd cycles (2k+1-cycles) for k=1,2. Moreover, a certain class of trees with maximum vertex degree Δ(T) ≤ 4 decreased computation time and enhances separation ability. Furthermore, calculation time was reduced by symmetry breaking constraints eliminating all but a few representative embeddings from each equivalence class of symmetrical embeddings. For this, grid symmetries due to translation, rotation and reflection of the embeddings were considered as well as vertex subsets whose inner permutations didn't change the value of the objective function.

4.4 Pattern recognition of gene regulation on the metabolic network

Neighboring enzymes on the adjacency matrices were grouped by combining their gene expression values with wavelet transforms. These transforms yielded combined expression values ("features") of low pass filters to detect similar expression changes and high pass filters to detect contrasting regulation patterns. The discriminative behavior of all non-trivial features was tested using Wilcoxon rank tests. Pathways were ranked according to their best discriminating features. Features were regarded as significant if they had p-values ≤ 0.05 after correction for multiple testing using the Bonferroni method (Bonferroni 1935, Gordi & Khamis 2004). Only pathways with more than three significantly, differentially regulated reactions and genes were further investigated to focus on the most relevant features. Two reactions that consisted of exactly the same genes were counted as one reaction.

Acknowledgements

This work was funded within the BMBF-FORSYS consortium Viroquant (#0313923), the Helmholtz Alliance on Systems Biology and the *Nationales Genom-Forschungs-Netz* (NGFN+) for the neuroblastoma project, ENGINE.

References

[Altenberg and Greulich, 2004] B. Altenberg , Greulich K.O. *Genes of Glycolysis Are Ubiquitously Overexpressed in 24 Cancer Classes***.** Genomics, 84:1014-1020, 2004.

[Chen, 2007] L. Chen, Pankiewicz, K. W. *Recent Development of Imp Dehydrogenase Inhibitors for the Treatment of Cancer***.** Curr Opim Drug Discov Devel 10:403-412, 2007.

[Chuang*, et al.*, 2007] H. Y. Chuang, Lee E., Liu Y. T., Lee D., Ideker T. *Network-Based Classification of Breast Cancer Metastasis***.** Molecular systems biology, 3:140, 2007.

[Debruyne*, et al.*, 2001] P. R. Debruyne, Bruyneel E. A., Li X., Zimber A., Gespach C., Mareel M. M. *The Role of Bile Acids in Carcinogenesis***.** Mutation research, 480-481:359-369, 2001.

[Ergun*, et al.*, 2007] A. Ergun, Lawrence C. A., Kohanski M. A., Brennan T. A., Collins J. J. *A Network Biology Approach to Prostate Cancer***.** Molecular systems biology, 3:82, 2007.

[Fan and Perou, 2006] C. Fan, Oh, D. S., Wessels, L., Weigelt, B., Nuyten, D. S. A., Nobel, A. B., Van't Veer, L. J., Perou C. M. *Concordance among Gene-Expression-Based Predictors for Breast Cancer***.** The New England Journal of Medicine, 355:560-569, 2006.

[Garewal*, et al.*, 1996] H. Garewal, Bernstein H., Bernstein C., Sampliner R., Payne C. *Reduced Bile Acid-Induced Apoptosis In "Normal" Colorectal Mucosa: A Potential Biological Marker for Cancer Risk***.** Cancer research, 56:1480-1483, 1996.

[Goeman*, et al.*, 2004] J. J. Goeman, Van De Geer S. A., De Kort F., Van Houwelingen H. C. *A Global Test for Groups of Genes: Testing Association with a Clinical Outcome***.** Bioinformatics, 20:93-99, 2004.

[Harris*, et al.*, 2004] M. A. Harris, Clark J., Ireland A., Lomax J., Ashburner M., Foulger R., Eilbeck K., Lewis S., Marshall B., Mungall C., Richter J., Rubin G. M., Blake J. A., Bult C., Dolan M., Drabkin H., Eppig J. T., Hill D. P., Ni L., Ringwald M., Balakrishnan R., Cherry J. M., Christie K. R., Costanzo M. C., Dwight S. S., Engel S., Fisk D. G., Hirschman J. E., Hong E. L., Nash R. S., Sethuraman A., Theesfeld C. L., Botstein D., Dolinski K., Feierbach B., Berardini T., Mundodi S., Rhee S. Y., Apweiler R., Barrell D., Camon E., Dimmer E., Lee V., Chisholm R., Gaudet P., Kibbe W., Kishore R., Schwarz E. M., Sternberg P., Gwinn M., Hannick L., Wortman J., Berriman M., Wood V., De La Cruz N., Tonellato P., Jaiswal P., Seigfried T., White R. *The Gene Ontology (Go) Database and Informatics Resource***.** Nucleic Acids Res, 32:D258-261, 2004.

[Kanehisa*, et al.*, 2008] M. Kanehisa, Araki M., Goto S., Hattori M., Hirakawa M., Itoh M., Katayama T., Kawashima S., Okuda S., Tokimatsu T., Yamanishi Y. *Kegg for Linking Genomes to Life and the Environment***.** Nucleic Acids Res, 36:D480-484, 2008.

[Katona and Stenson, 2009] B. W. Katona, Anant, S., Covey, D. F., Stenson W. F. *Characterization of Enantiomeric Bile Acid-Induced Apoptosis in Colon Cancer Cell Lines***.** The journal of biological chemistry 284:3354-3364, 2009.

[Li, 1993] Y. Li. *Oxaloacetate Induces DNA Synthesis and Mitosis in Primary Cultured Rat Hepatocytes in the Basence of Egf* Biochemical and Biophysical Research communications, 193:1339-1346, 1993.

[Lui, 1982] M. S. Lui. *Biochemical Pharmacology of Acivicin in Rat Hepatoma Cells***.** Biochem Pharamcol 31:3469-3473, 1982.

[Mootha*, et al.*, 2003] V. K. Mootha, Lindgren C. M., Eriksson K. F., Subramanian A., Sihag S., Lehar J., Puigserver P., Carlsson E., Ridderstrale M., Laurila E., Houstis N., Daly M. J., Patterson N., Mesirov J. P., Golub T. R., Tamayo P., Spiegelman B., Lander E. S., Hirschhorn J. N., Altshuler D., Groop L. C. *Pgc-1alpha-Responsive Genes Involved in Oxidative Phosphorylation Are Coordinately Downregulated in Human Diabetes***.** Nat Genet, 34:267-273, 2003.

[Nacu*, et al.*, 2007] S. Nacu, Critchley-Thorne R., Lee P., Holmes S. *Gene Expression Network Analysis, and Applications to Immunology***.** Bioinformatics (Oxford, England), 2007.

[Nemhauser and Wolsey, 1999] Gl. Nemhauser, Wolsey La.: *Integer and Combinatorial Optimization.* John Wiley & Sons 1999.

[Oakman and Di Leo, 2009] C. Oakman, Bessi, S., Zafarana, E., Galardi, F., Biganzoli, L., Di Leo A. . *New Diagnostics and Biological Predictors of Outcome in Early Breast Cancer***.** Breast Cancer Research 205:1-11, 2009.

[Pai*, et al.*, 2004] R. Pai, Tarnawski A. S., Tran T. *Deoxycholic Acid Activates Beta-Catenin Signaling Pathway and Increases Colon Cell Cancer Growth and Invasiveness***.** Molecular biology of the cell, 15:2156-2163, 2004.

[Patil and Nielsen, 2005] K. R. Patil, Nielsen J. *Uncovering Transcriptional Regulation of Metabolism by Using Metabolic Network Topology***.** Proc Natl Acad Sci U S A, 102:2685-2689, 2005.

[Pedersen-Bjergaard, 1985] J. Pedersen-Bjergaard. *Risk of Acute Nonlymphocytic Leukemia and Preleukemia in Patients Treated with Cyclophosphamide for Non-Hodgkin's Lymphomas. Comparison with Results Obtained in Patients Treated for Hodgkin's Disease and Ovarian Carcinoma with Other Alkylating Agents* Ann Intern Med 103:195-200, 1985.

[Pitman, 1937] E. J. G. Pitman. *Significance Tests Which May Be Applied to Samples from Any Population***.** Royal Statistical Society Supplement, 4:119-130, 225-232, 1937.

[Spychala, 2003] J.: Spychala. *Regulation and Function of Ecto-5'-Nucleotidase and Adenosine in Cancer* 39 th Meeting of the Polish Biochemical society 185, 2003.

[Suzuki and Sasano, 2005] T. Suzuki, Miki, Y., Nakamura, Y., Moriya, T., Ito, K., Ohuchi, N., Sasano H. *Sex Steroid-Producing Enzymes in Human Breast Cancer***.** Endocrine-Related Cancer, 12:701-720, 2005.

[Van 'T Veer*, et al.*, 2002] L. J. Van 'T Veer, Dai H., Van De Vijver M. J., He Y. D., Hart A. A., Mao M., Peterse H. L., Van Der Kooy K., Marton M. J., Witteveen A. T., Schreiber G. J., Kerkhoven R. M., Roberts C., Linsley P. S., Bernards R., Friend S. H. *Gene Expression Profiling Predicts Clinical Outcome of Breast Cancer***.** Nature, 415:530-536, 2002.

[Van De Vijver and Bernards, 2002] M. J. Van De Vijver, He, Y. D., Van't Veer, L. J., Dai, H., Hart, A. A. M., Voskuil, D. W., Schreiber, G. J., Peterse, J. L., Roberts, C., Marton, M. J., Parrish, M. , Atsma, D., Witteveen, A., Glas, A., Delahaye, L., Van Der Velde, T., Bartelink, H., Rodenhuis, S., Rutgers, E. T., Friend, S. H., Bernards R. *A Gene-Expression Signature as a Predictor of Survival in Breast Cancer***.** The New England Journal of Medicine, 347:1999-2009, 2002.

[Xu and Huang, 2005] R. Xu, Pelicano, H., Zhou, Y., Carew, J. S., Feng, L., Bhalla, K. N., Keating, M. J., Huang P. *Inhibition of Glykolysis in Cancer Cells: A Novel Strategy to Overcome Drug Resistance Associated with Mitochondrial Respiratory Defect and Hypoxia***.** Cancer Research, 65:613-621, 2005.

[Yan and Sun, 2008] X. Yan, Sun F. *Testing Gene Set Enrichments for Subset of Genes: Sub-Gse***.** BMC Bioinformatics 9:1-15, 2008.

[Zimber*, et al.*, 1994] A. Zimber, Chedeville A., Gespach C., Abita J. P. *Inhibition of Proliferation and Induction of Monocytic Differentiation on Hl60 Human Promyelocytic Leukemia Cells Treated with Bile Acids in Vitro***.** International journal of cancer, 59:71-77, 1994.

[Zimber and Gespach, 2008] A. Zimber, Gespach C. *Bile Acids and Derivatives, Their Nuclear Receptors Fxr, Pxr and Ligands: Role in Health and Disease and Their Therapeutic Potential***.** Anti-cancer agents in medicinal chemistry, 8:540-563, 2008.

Fast and Scalable Pattern Mining for Media-Type Focused Crawling*

[experience paper]

Jürgen Umbrich and **Marcel Karnstedt** and **Andreas Harth**†
Digital Enterprise Research Institute (DERI)
National University of Ireland, Galway, Ireland
firstname.lastname@deri.org

Abstract

Search engines targeting content other than hypertext documents require a crawler that discovers resources identifying files of certain media types. Naïve crawling approaches do not guarantee a sufficient supply of new URIs (Uniform Resource Identifiers) to visit; effective and scalable mechanisms for discovering and crawling targeted resources are needed. One promising approach is to use data mining techniques to identify the media type of a resource without the need for downloading the content of the resource. The idea is to use a learning approach on features derived from patterns occuring in the resource identifier. We present a focused crawler as a use case for fast and scalable data mining and discuss classification and pattern mining techniques suited for selecting resources satisfying specified media types. We show that we can process an average of 17,000 URIs/second and still detect the media type of resources with a precision of more than 80% and a recall of over 65% for all media types.

1 Introduction

The number of search engines focused on specific topics has increased significantly over recent years. Besides search engines focused on (hyper)text documents, specialised search engines are available online which collect and integrate information from files of particular media types. Seeqpod [URL, j] and Blinkx [URL, g] offer search over audio and video files, Google Scholar [URL, b] and CiteSeer [URL, a] are digital libraries of printable documents, Technorati [URL, d] provides real-time access to news-feeds, and Seekda [URL, c] offers search capabilities for web services. A common issue for all vertical search engines is the challenge of discovering and downloading the targeted files on the Web. Specifically, the challenge of detecting documents of a certain media type without inspecting the content is still not solved [Lausen and Haselwanter, 2007]. For this task, a URI-only classifier is a good choice, because speed is crucial and content filtering should be enabled before an (objectionable) web page is downloaded. Basically, a focused crawler ([Chakrabarti *et al.*, 1999]) wants to infer the topic of a target page before devoting bandwidth to download it. Further, a page's content may be hidden in images.

A crawler for media type targeted search engines is focused on the document formats (such as audio and video) instead of the topic covered by the documents. For a scalable media type focused crawler it is absolutely essential to discover documents of the requested media type on the Web and to avoid expensive HTTP lookups of irrelevant files. Thus, the crawler needs to identify the media type of a document without establishing a connection and downloading the content. A common way to identify the format of a file is to use the file extension of the file name or to detect characteristic byte patterns in the file content itself (magic number approach), which does not scale well. The latter approach is not suitable because it requires to retrieve the data which is expensive and time consuming task. We can conclude that the file extension is only for some media types suitable as an identifier based on a study of 22M Web documents [Umbrich *et al.*, 2008] in 2008.

We propose to use classification or pattern mining techniques to discover Web documents of requested media types without analysing the content. For this, we utilise information available for a crawler during the crawling process to classify and filter URIs for pre-defined media types. Note that learning patterns without analysing the content of files can be applied in other scenarios as well, e.g., in genre classification. We present general data mining approaches suited for this task, discuss their strengths and weaknesses, and, based on first experiences, present a classifier-based solution. As this method still comes with several disadvantages, we further propose to apply frequent pattern mining approaches as an alternative. Here we focus on stream mining approaches, as they provide a fast and scalable solution that is perfectly suited for the long-running and input-intensive task of a Web crawler.

The remainder of the paper is organised as follows: In Section 2 we briefly present the basics of a crawler focusing on media-types and discuss general data mining approaches suited for this. Section 3 presents first experiences that we gained using implementations available in the WEKA toolkit. Based on these experiences, we propose a classifier approach in Section 4 and discuss an improvement based on pattern stream mining. Section 5 contains an evaluation of the classifier approach. Finally, Section 6 briefly presents related work and Section 7 concludes the paper.

*This material is based upon works jointly supported by the Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2) and under Grant No. 08/SRC/I1407 (Clique: Graph & Network Analysis Cluster)

†Current affiliation: Institut AIFB, Universität Karlsruhe (TH)

2 Data Mining for Focused Crawling

The principle of a crawler is to start with a set of seed URIs and to recursively follow the discovered links. For this, crawlers follow, for instance, a breadth-first or depth-first approach. This means, the crawler changes domains through the crawling job and may be returning to domains already visited before. A focused crawler tries to follow only links to pages and files of a specific type – in our case the media type of files.

Our basic idea is to utilise the information available for a crawling during the crawl loop to identify the media-type of Web resources without inspecting the content itself. The accessible information sources for a crawler are [Umbrich *et al.*, 2008]:

1. the **URI**
2. the **link position** of the URI in the HTML document
3. the information sent with the **HTTP response header**

However, in this work we exclusively focus only on the mining of features contained in the URI. URIs are also used for topic classification [Baykan *et al.*, 2009]. Media types are registered with their related RFC description with IANA [URL, h]. The RFC description contains a general explanation of the purpose of the media type and also recommendation for the file extension(s) to use when publishing a document. A media type consists of two parts, 1) the `content-type` and 2) the `sub-type`, separated by "/" (`content-type`/`sub-type`). To explain our thoughts we introduce an example URI[1] referring to a W3C Video on the Web Workshop in 2007.

2.1 URI

Every network-retrievable document has a Uniform Resource Identifier (URI) as defined in RFC2396 [URL, i], which specifies that a URI consists of five components.

[*PROTOCOL*]://[*HOST*]:[*PORT*]/[*PATH*][*FILE*]?[*QUERY*]

Please note that we focus exclusively on the Hypertext Transfer Protocol (HTTP) and omit the other protocols, such as the File Transfer Protocol (FTP, RFC 959 [URL, f]) or the gopher protocol (RFC 1436 [URL, e]). The single parts have the following meanings:

- The *PROTOCOL* specifies the transfer protocol used to access the resource on the Internet.
- The *HOST* part refers to the domain name of the web server.
- The *PORT* component depends on factors such as server-side firewall settings, proxy configuration or router settings. If the *PORT* component is omitted, the standard assumes the default HTTP port 80.
- The *PATH* and *FILE* components can be created either automatically, for example by a content management system, or manually by human users.

The *PATH* component can be part of a hierarchical folder structure. Users organise their content in a folder structure, such as storing images in an `image` directory or videos in a `video` sub folder. In this case, special sub folders can be used as an indicator for the media types of the documents in this folder. Our URI example has the *FILE* component `Angio.avi` with file extension `avi`, which is mapped to the media type `video/x-msvideo`. The mapped media type matches with the real media type of our example. The *QUERY* component is a string of information to be interpreted by the resource and is omitted here. We use the following notation for the URI components in this paper:

- **E** for the file extension of the *FILE* component
- **F** for the filename
- the single tokens in the *PATH* components are labeled with **T**
- the *HOST* part is labeled with **D**

2.2 Classification

We can map our media-type identification problem to a classification task. Based on a set of predefined media types, we want to classify a URI to its real media type. Therefore, we have to split the components of a URI into a feature set that serves as the input for the classification algorithm. Possible classification algorithms for this task could be one of the following: The **Bayesian classifier** shows convincing results for classifying text. Most spam detection applications use a Bayesian classifier to decide whether or not a text snippet or an email is spam[2] [Sahami *et al.*, 1998]. The **support vector machines** (SVM) algorithm is also used for focused web crawling [Sizov *et al.*, 2002] and achieved reasonably good results in classifying documents to topics. The C4.5. algorithm generates a pruned or **unpruned decision tree** [Quinlan, 1993] [Quinlan, 1996]. A common **Bayesian network** classification algorithm [Pearl, 1985] is also featured. Noteworthy, it is proven that a classification algorithm based on decision trees and word tokens from anchor text can significantly improve focused crawling [Li *et al.*, 2005]. Hence, we do not know how it performs for the classification task based on a small set of features.

Some disadvantages come along with the classification approach. If we use a supervised algorithm we need to generate a training set. Given that there exist over 200 different media types on the Web and that 80% of the documents are text/html documents, we have the problem of gathering a training set with representative candidates. Another important fact is the dynamic and heterogeneous nature of the Web. Different domains use different URI patterns for the same media type documents. The classifier could be optimised especially for some certain domains, depending on the training set of selected URIs. Such a domain specific classifier could achieve a very good classification precision for certain URIs, but the classifier will "fail" for URIs of domains that are not in the training set.

This static approach is not suitable without retraining the classifier during the crawl. This is especially an issue in the context of changing domains, where already visited domains may be visited later on again. Also, the training and testing tasks could be very time consuming in the context the Web crawling application. We will show that in the next section.

2.3 Dynamic Pattern Mining

A very promising approach is the mining of patterns from URIs to identify the media type of the underlying document. Especially stream pattern mining seems to be a good solution. We will discuss the stream mining approach in more detail in Section 4. The basic idea is to mine frequent

[1] http://www.w3.org/2007/08/video/slides/KidsHealth/Angio.avi

[2] http://www.paulgraham.com/Spam.html

Content type	Classifier	Max precision	Features
application/*	j48	95.24%	DTFE
audio/*	Bayes	94.09%	TE
image/*	smo	100.00%	E
text/*	smo	91.03%	DTFE
video/*	Bayesnet	56.96%	TE

Table 1: Selected results for the best classifier & feature combination with the highest precision for each tested content types.

patterns from a set of URIs and their real media types. Based on these association patterns we can learn rules and use them to identify the media types. With pattern stream mining approaches we can discover new patterns over the time the crawler traverses the Web. This approach can also keep up with new appearing media types and does not rely on a predefined and static set of media types.

3 First Experiences

First, we wanted to know if a classification approach is suited for our needs at all. We used the WEKA toolkit to gain first knowledge if patterns extracted from URIs are suitable for a media type classification. Based on the possible features of a URI, we tested and compared four different classification algorithms provided by the WEKA framework. The evaluation contains the results of all possible permutations of classifier and feature combination for media types and content types. However, we present here only selected results out of 19 test runs. The evaluation shows precision and recall values as well as detailed results of the time needed to train and test the classifiers. Further, we studied the scalability of the WEKA library and the provided implementations.

3.1 Feature Combinations

The test for possible feature combinations contains the following four feature combinations: **E**, **FE**, **TFE**, **DTFE**. They were chosen intuitively in order to get a first impression of the suitability of the approach. We observed that there exists no clear feature combination that in general achieves the highest precision and recall values. In 9 out of 19 tests the feature combination DTFE (domain, tokens, file name and file extension) achieved the best precision. Another objective fact is that there exists no single classification algorithm that clearly outperforms the others. Table 1 shows the summary of the classifiers that have the highest precision to identify documents for the tested media types. 12 out of 19 classifier algorithms uses the Bayesian theory. The Bayes classifier achieves in 8 out of 19 cases the best precision value and the Bayesian net algorithm in another 4 cases. The complete list of the results[3] and more details are provided in [Umbrich, 2008].

3.2 Scalability

We stopped the execution time to train and test a certain WEKA classifier after a number of input instances. The benchmarks are performed for all possible combinations of classifiers and feature patterns for the content type `application` with a dictionary containing 5,000 words. The results are for the feature combination DFTE. Figure 1 shows the time to generate the WEKA input (ARFF) file, Figure 2 shows the result to train the algorithms and Figure 3 the results to test the trained classifiers. The major problem and scalability limitation of the WEKA workbench is that all the information that is needed to train a classifier and finally to classify new instances are kept in memory. Thus, the limitation of WEKA is the available in-memory space. With the naive Bayes classifier the memory limit was reached with a dictionary containing 20,000 tokens. Another reason why we believe that WEKA is not suitable for a scalable architecture is the bad time performance to classify a list of URIs. Results show that the average classification speed is 47.5 URIs/seconds. Under the assumption that the system filters millions of URIs, the current implementation with the WEKA libraries is not applicable. The time to create the required ARFF file increases exponentially with the number of instances, as the results in Figure 1 show.

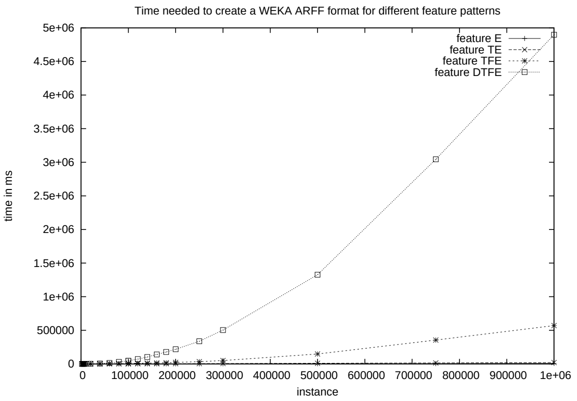


Figure 1: Elapsed time to generate the WEKA ARFF input format.

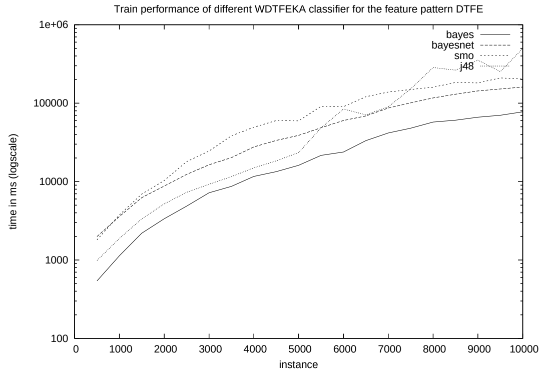


Figure 2: Elapsed time to train a WEKA classifier for each feature combination.

The conclusion of this first evaluation is that the provided JAVA implementations of WEKA have memory limitations. For our Web crawler application, the classifiers cannot keep up with the performance of URIs/sec supplied by the crawler.

4 Approach

From the tested classification algorithms, the Bayesian classifier was the most suitable, in terms of used mem-

[3] http://www.umbrich.net/pubs/master_thesis.pdf

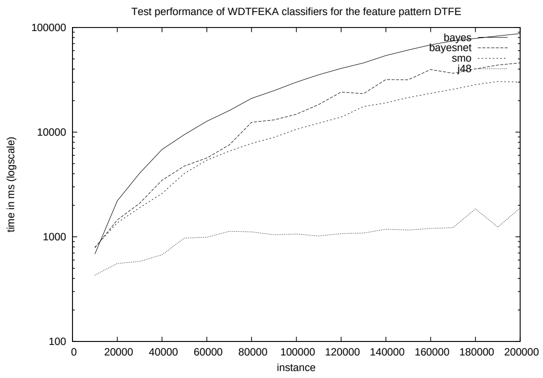


Figure 3: Elapsed time to test a WEKA classifier for each feature combination.

ory and the performance of the learning and classification task. Thus, we propose a statistical classifier similar to the *a priori* approach of the Bayesian algorithm or "One-Item-Rule" of an association rule algorithm [Agrawal *et al.*, 1993].

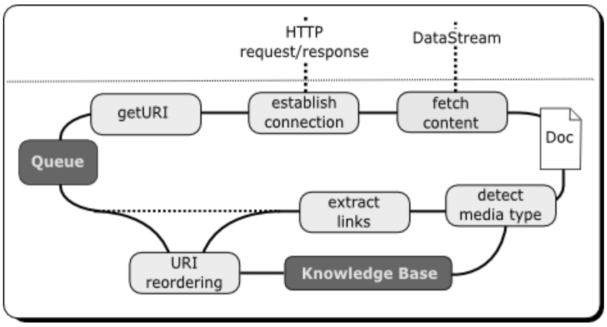


Figure 4: Media type crawl filter architecture

The general idea is illustrated in Figure 4. A knowledge base holds the data to generate files required to build and evaluate new classifiers. Furthermore, the knowledge base analyses the inserted information and generates meta data about, for example, the distribution of media types, the occurrence of feature values and the appearance of features together with media types. These meta data are used for a statistical approach to determine the media type of a URI.

An evaluation function calculates a conditional probability of being a media type mt, given the features extracted from URI uri. In the current version, we implemented a media type filter that supports a statistical classifier. This is an algorithm similar to association rule mining, which is fed by background knowledge from the meta data of the data store component. Next, we will describe the algorithm of the proposed statistical classifier in detail.

4.1 Statistical Classifier

The statistical classifier calculates the relevance score based on the conditional probability with the function of the Bayesian theory,

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{|A \cap B|}{|B|}.$$

The classifier scans the meta information of the data store to select the best features to identify the media types and stores the selected features in an internal cache. The most precise features are selected with an algorithm similar to the *a priori* association rule mining algorithm [Agrawal *et al.*, 1993; Agrawal and Srikant, 1994]. The "One-Item-Rule"($A \Rightarrow B$) algorithm selects single features which are associated with a "goal" or class like $.jpg \Rightarrow image/jpeg$. The best features are selected based on the confidence value

$$conf(a \rightarrow b) = \frac{sup(a \cup b)}{sup(a)}$$

with

$$sup(x) = \frac{\#x}{\#total}.$$

The formula for calculating the confidence is equivalent to the conditional probability formula, which calculates the probability of some event A, given the occurrence of some other event B. Conditional probability is written

$$P(A|B) = \frac{P(A \cup B)}{P(B)},$$

and is read "the probability of A, given B".

The algorithm is not a full association rule algorithm and skips the rule detection for all possible feature combination. Instead, our algorithm uses the mapping function

$$\Pi(uri, featurePattern) = featureString$$

to generate all possible combination of features for given and pre-defined combination patterns.

A combination pattern is a series of single features, such as the combinations of **DT**, that is, all possible combination of the pair <pld+token>.

We use a true Bayesian estimate to calculate the importance of a rule based on the value of the conditional probability and the occurrence of the feature in our data store.

Bayesian Estimate The formula for calculating the weighted score for a "feature rule" gives a true Bayesian estimate:

$$wfeature = \frac{v}{v+m} \times R + \frac{m}{v+m} \times C$$

where R denotes probability for the feature, v the number of total occurrence of the feature, m the minimum occurrence required to be added to the rule set and C the default probability. This equation is adapted from the weighted rank of movies at IMDB[4].

The media type filter is the core component of our crawling framework to guide the crawler while traversing the Web for new documents of requested media types. With this component, the framework can perform a crawling strategy which is equal to a dynamical heuristic search algorithm. The media type filter integrates the mapping function $\Pi(uri)$ with an evaluation function $f(uri, \Pi(uri))$. The background knowledge used to calculate the relevance score of URIs can be updated during the crawl runtime and thus, the filter can learn new features to discover relevant documents. The mapping function $\Pi(uri) = \{TLD(uri), T(uri), N(uri), E(uri)\}$ extracts features from a URI and generates a feature vector that is used as input for the evaluation function. The evaluation function returns the conditional probability that a URI

[4] http://www.imdb.com/

is of media type mt, or expressed in terms of the probability theory $f(mt, \Pi(uri)) = P(mt|uri)$. With the relevance score from the evaluation function, the component filters a list of URIs and adds selected URIs to a single priority queue or a set of priority queues. The background knowledge of this component contains pairs of URIs with their real media types. The post-processing component detects the real media type of downloaded files and adds this knowledge into the media-type filter.

Filter rules control which pages are visited next and which URIs are ignored. The crawling behavior can be controlled with the following rules:

- $(mt, min\_prob) \Rightarrow ALLOW$ Allow/Select URIs with a probability of more than min_prob for media type mt.
- $(mt, min\_prob) \Rightarrow DISALLOW$ Disallow/Filter out URIs with a probability of more than min_prob for media type mt.

We are aware that the presented approach of a statistical classifier does not allow to easily change patterns or learn new media types. Obviously, an unsupervised approach that can mine the occurring patterns would be more practical. Hence, we discuss a pattern mining approach in the next section.

4.2 Pattern Tree Approaches

As we highlighted, the **limitations** of the presented approach are:

1. feature patterns are static and pre-defined
2. more feature pattern combinations require more in-memory space
3. new discovered media types cannot be added to the classifier

We found a promising approach using frequent pattern trees [Han *et al.*, 2000] and an extension to mine frequent itemsets in streams under dynamically changing resource constraints [Franke *et al.*, 2006]. Especially, the latter work fulfills our requirements. Franke et. al provide a framework to mine frequent itemsets, either from fixed size intervals or from time intervals. The algorithm is resource-aware, which means it can be adjusted to changing resources and it can be run with fixed allocated resources. This approach has the following **advantages**:

1. automatic detection of new patterns
2. assurance of a constant memory footprint independent from the crawling time or the number of processed URIs.
3. new discovered media types can be added to the classifier

Using a stream-based approach promises that the performance requirements of a Web crawler can be met without problems. The crawler produces a high input rate for the mining task (up to thousands of inserts per minute) and runs for days to weeks. Thus, we are running continuous mining queries on a continuous stream of high input rate. That is exactly what stream mining algorithms are developed for, as they usually aim for needing only one look on each date. Another advantage is the possibility to query for frequent itemsets from certain time intervals. If the crawler records at which time he enters or leaves a domain, it is straightforward to determine patterns from only that domain. In future work we plan to evaluate what is the better choice, either use all patterns learned so far or just focus on domain-specific ones. The algorithm from [Franke *et al.*, 2006] provides a perfect basis for that, which is accompanied by its resource awareness. Moreover, the algorithm can always provide guarantees on the achieved quality, depending on the currently used resources.

Meanwhile, we implemented the pattern mining approach. The next steps are to fully integrate it into the crawling framework. When this is done, we can measure the overall performance of this method and compare it to the static classifier approach. We expect similar performance, due to the stream character of the used solution. Concerning accuracy and applicability, we expect even better results, due to the dynamic and unsupervised features of the approach.

5 Evaluation

In this section, we present the methods and results of our evaluation of the statistical classifier. We measured the precision and recall of our implementation and further, the performance with respect to the requirements of a Web crawler use-case. We measured the discovery ratio of requested media types on the Web based on a base-line crawl with a breadth-first crawling strategy. First, we present the evaluation setup and used methods, followed by the results of the single evaluations steps. Finally, we provide a detailed discussion of the results.

5.1 Setup

All experiments are performed on a single Opteron 2.2 GHz CPU with 4 GB of main memory and two SATA (160GB,750GB) disks. We used a published and representative web corpus containing 22.2M documents [Umbrich *et al.*, 2008]. The test corpus contains 3.8M external links from ODP[5] and 6.4m external links from Wikipedia[6] in December 2007.

We use data from this combined corpus to evaluate the approach of our statistical classifier. We focus on the DTFE feature combination, as this has been proven to be the generally most suited one in our WEKA tests. The underlying assumption for this test is that as soon as the conditional probability of a feature is greater than 0.5 (=50%), we can assume a positive match. The classifier evaluates first the file extension of the URI, second the single path tokens and third, the combination of the top-level-domain and the path tokens. If all features cause a relevance value of less than 0.5, we will assume a false match.

Our test set contains 4.28 M detected media types and is split into eight folds, fold `A`, `B`, `C`, `D`, `E`, `F`, `G` and `H`. The last fold (H with 285396 entries) is used to evaluate and test the classifier. The first seven folds (`A-G`), containing 500.000 entries, are used to train and update the statistical classifier. We ran seven evaluation rounds for each of the five content types (`application`, `audio`, `image`, `text` and `video`). The classifier is updated by one of the remaining data set parts and tested against the separate data set `H` in each round. The internal cache of the classifier was set up with 5000 entries for each feature and with the thresholds 0.5 and 0.7. All features that have a conditional probability less than the threshold are deleted.

[5] http://rdf.dmoz.org/rdf/content.rdf.u8.gz

[6] http://download.wikimedia.org/enwiki/

5.2 Results

First, we list the precision and recall values from the evaluation. Second, we show the benchmarks to mine and classify new URIs. Table 2 list the precision and recall values for two thresholds and the five content types.

	Rounds						
	a	b	c	d	e	f	g
application/*							
Threshold 0.5							
Pr	85.24	85.34	86.21	85.93	86.05	85.84	85.67
Re	66.56	67.37	67.54	67.68	67.50	67.46	67.47
Threshold 0.7							
Pr:	91.98	91.96	93.34	93.10	93.22	93.04	93.13
Re:	66.50	68.44	66.48	65.38	65.34	65.32	65.32
audio/*							
Threshold 0.5							
Pr:	87.88	86.14	86.57	86.57	84.88	86.57	84.69
Re:	84.88	84.88	84.88	84.88	84.88	84.88	86.34
Threshold 0.7							
Pr:	91.58	91.58	91.10	91.10	91.10	91.10	91.10
Re:	84.88	84.88	84.88	84.88	84.88	84.88	84.88
image/*							
Threshold 0.5							
Pr:	84.89	84.05	83.50	82.96	83.37	83.53	83.23
Re:	98.31	98.43	98.64	98.64	98.66	98.66	98.68
Threshold 0.7							
Pr:	87.84	87.81	87.90	87.74	87.89	87.60	87.66
Re:	98.11	98.25	98.37	98.46	98.52	98.50	98.50
text/*							
Threshold 0.5							
Pr:	73.68	73.78	73.81	74.00	73.94	73.97	73.99
Re:	99.78	99.77	99.74	99.73	99.74	99.75	99.74
Threshold 0.7							
Pr:	86.74	87.26	87.47	88.85	88.71	88.77	88.83
Re:	61.36	61.07	61.01	60.06	60.16	59.91	59.90
video/*							
Threshold 0.5							
Pr:	72.73	72.73	71.11	72.73	71.11	71.11	71.11
Re:	91.43	91.43	91.43	91.43	91.43	91.43	91.43
Threshold 0.7							
Pr:	81.48	81.48	74.42	81.48	74.42	74.42	74.42
Re:	62.86	62.86	91.43	62.86	91.43	91.43	91.43

all values are percentages.

Table 2: Precision and recall of the statistical classifier for different thresholds.

Scalability We benchmarked the implementation of the statistical classifier to measure the time to insert/update and classify a list of URIs. Figure 5 shows results. Our implementation shows a linear increase of the elapsed time with the number of URIs for both operations.

Focused Crawling for Media Types We evaluate the applicability of our approach in a media-type focused Web crawler based on the following use case: Documents of content type audio, video or image are requested to build a multimedia search engine. The detailed setup of this experiment is as follows: We started with a seed set of twelve URIs, collected from the social bookmark page Delicious[7]. The requested documents cannot be used to extract new links, thus we selected also links to text/html documents. To measure the efficiency of the pattern mining approach we compared three typical crawling strategies. As our base-line we performed a breadth-first crawl. The second strategy is a focused crawl (best-first) using the statistical classifier without a cut-off threshold to omit URIs. We prioritised URIs based on their conditional probability that they are of a requested media type. The filter methods first selects all URIs which are of content type audio,video or image and adds the selected URIs to the queue. In a second processing step URIs of media type text/html are selected and forwarded to the queue. The third strategy applies a fixed size queue of 10K URIs and a URI-per-domain limit of 10. The results of the different strategies are presented in Figure 6.

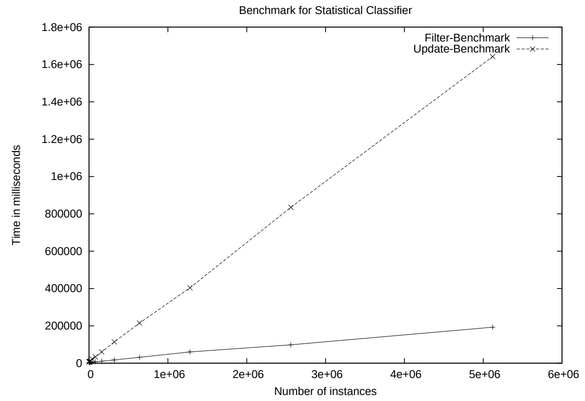


Figure 5: Performance of the statistical classifier.

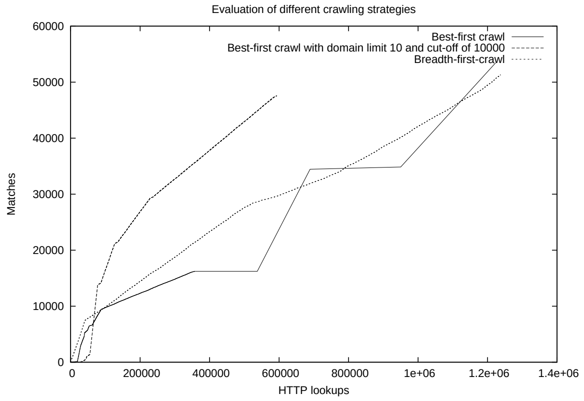


Figure 6: Performance of different crawl strategies.

5.3 Discussion

Next, we will discuss the results presented in the previous section. Because of space limitations we refer to a more detailed discussion of the results in [Umbrich, 2008].

Precision and Recall Table 2 lists the detailed results for the classification evaluation. First of all, we can observe that for all tested content-types the precision values are higher than 75% and the recall values above 60% . The predefined feature patterns achieve reasonable good results. However, we observe that the recall values for video/* and threshold 0.7 differ between 60% and 90%. This difference is caused by the pre-defined and static feature pat-

[7] http://delicious.com/

terns and boosts our idea to use the frequent pattern tree approach.

Furthermore, we observe an increase of the precision by nearly 9% if the internal cache threshold of the classifier is changed from 0.5 (= 50%) to 0.7 (=70%). To revive, features with a conditional probability less than the internal cache threshold for a certain content or media type are not recorded and used for the classification. With this threshold we can control the characteristics of the statistical classifier. A higher threshold results in a higher precision and a lower threshold increases the recall.

Comparing the precision of the statistical classifier with the achieved classification precisions of the WEKA classifiers we notice very promising results. Beside the precision values for the classification of documents of content type `image`, the statistical classification approach achieves precision values that are only around 2% less than the WEKA classifiers, with still reasonable good recall values for the content types `application`, `audio`, `text` and `video`.

As expected, the classifier approach achieves very good values for precision and recall. This shows that the approach of choosing data mining techniques to identify media types without checking file content is very suitable and applicable. The next interesting step is to evaluate the dynamic pattern mining approach, which promises to be even better suited, due to its streaming and unsupervised character.

Fast and Scalable The processing speed of the current implementation of the statistical classifier is in average 17,385 URIs/second for filtering URIs and in average 3,000 URIs/second for inserts and updates of new obtained information. The performance clearly outperforms the processing speed of the WEKA classifiers. For filtering single URIs our implementation is 319 times faster than the algorithm using the WEKA classifiers and 15 times faster compared to the best achievable processing speed of WEKA (generating an ARFF input file with all URIs ⇒ 200 URIs/sec). The performance for training the classifier is in average 27 times faster then the best possible performance of the WEKA implementation (110 URIs/second for the Bayesian classifier) for the same feature combinations.

The evaluation of the focused crawling strategy implemented in the current version of our crawler shows that the crawler continuously gathers the requested documents of content types `image`, `audio` and `video` with the breadth-first crawling strategy. The gradient of the number of fetched relevant documents depends only on the available requested documents in the crawl queue. We can see that the gradient of the curve varies over time and is not constant. Depending on the URIs in the queue, the crawler extracts new links for a fraction of HTML documents that contain more relevant documents as the HTML document in the round before or reverse.

The curve of the näive breadth-first-crawl strategy shows a step function. This clearly shows the filtering and prioritising of URIs leading to relevant documents. In the very beginning of the crawl (<50,000 HTTP lookups) the number of the downloaded relevant documents is zero and then, suddenly, it significantly increases. The explanation for this is that the media-type filter is untrained in the beginning and cannot apply any background knowledge. With more and more visited documents and more obtained information of URIs and their belonging media types, the filter component discovers convenient feature patterns and is capable to identify and filter relevant documents.

The results show that even the a priori method used in combination with a focused crawler can meet the performance requirements of today's Web crawlers. However, performance of Web crawlers can never be good enough, as they usually represent extremely long-running tasks. That is why the stream-based pattern mining approach promises to be a very good choice. We expect it to at least meet the performance we gained in the evaluation presented here, if not even to be capable of producing better results.

6 Related Work

To the best of our knowledge, we are not aware of published work that focuses on the topics of focused crawling for certain media types beside the work of Bachlechner et. al. [Bachlechner *et al.*, 2006], which tries to gather "Web Service Description Language" (WSDL) files on the web. However, the work in hand focuses on the applicability and scalability of different data mining approaches for this task. [Baykan *et al.*, 2009] showed that patterns in URIs can be used for topic classification of the documents identified by the URIs. This is a similar approach to the one presented here. But, the application to media types presents specific requirements and challenges that are not discussed in the field of topic classification.

The list of used classifier algorithms in focused crawlers contains, among others, the application of: a simple naive bayes classifier [Passerini *et al.*, 2001], a k-nearest neighbour clustering algorithm [Ester *et al.*, 2004], a support vector machine [Sizov *et al.*, 2002], a decision tree [Najork and Wiener, 2001] [Li *et al.*, 2005], a neural network [Menczer *et al.*, 2001] and also a solution with hidden markov models [Liu *et al.*, 2004]. They all bear the disadvantage of requiring a supervised approach, similar to the classifier approach presented in this work. None of the works from above considers an unsupervised learning approach, neither they discuss the applicability of streammining techniques. Further, none of these works discusses media-type focused crawling.

7 Conclusion

Specialised search engines face major difficulties to discover and gather in a scalable and efficient way structured content of requested media types on the Web. Näive solutions such as gathering URIs via user submissions or crawling the entire Web (existing of 41 billion unique URIs with over 80% of `text/html` documents) for the targeted files do not guarantee a sufficient supply of URIs.

We investigated the problems of specialised search engines in discovering and gathering relevant documents from the Web. We first showed that classification approaches are suited in principle. Afterwards, we proposed an approach for scalable and optimised focused crawling that discovers URIs of targeted media types and extracts meta data in a structured format from the downloaded content. We were able to show that data mining techniques are well suited for implementing fast and scalable focused crawling. However, the choice of applied technique is a rather crucial one. Even if static classifier approaches work well and achieve good performance and accuracy, there is still great potential to increase both. We are eager to integrate and evaluate the pattern mining approach developed for data streams in order to judge on its applicability. We hope that we can increase performance and accuracy by this even more – while being able to adapt to the usually strict resource limitations that Web crawlers have to face.

References

[Agrawal and Srikant, 1994] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*, pages 487–499. Morgan Kaufmann, 1994.

[Agrawal *et al.*, 1993] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, 1993*, pages 207–216. ACM Press, 1993.

[Bachlechner *et al.*, 2006] Daniel Bachlechner, Katharina Siorpaes, Holger Lausen, and Dieter Fensel. Web service discovery - a reality check. In *Proceedings of the 1st Workshop: SemWiki2006 - From Wiki to Semantics, co-located ESWC*, Budva, Montenegro, June 2006.

[Baykan *et al.*, 2009] Eda Baykan, Monika Henzinger, Ludmila Marian, and Ingmar Weber. Purely url-based topic classification. In *18th International World Wide Web Conference*, pages 1109–1109, April 2009.

[Chakrabarti *et al.*, 1999] Soumen Chakrabarti, Martin van den Berg, and Byron Dom. Focused crawling: A new approach to topic-specific web resource discovery. *Computer Networks*, 31, 1999.

[Ester *et al.*, 2004] Martin Ester, Hans-Peter Kriegel, and Matthias Schubert. Accurate and efficient crawling for relevant websites. In *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases*, pages 396–407. VLDB Endowment, 2004.

[Franke *et al.*, 2006] C. Franke, M. Karnstedt, and K. Sattler. Mining data streams under dynamicly changing resource constraints. In *KDML 2006: Knowledge Discovery, Data Mining, and Machine Learning*, pages 262–269, October 2006.

[Han *et al.*, 2000] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *SIGMOD '00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 1–12, New York, NY, USA, 2000. ACM.

[Lausen and Haselwanter, 2007] Holger Lausen and Thomas Haselwanter. Finding web services. In *1st European Semantic Technology Conference*, volume 2007. ESTC, June 2007.

[Li *et al.*, 2005] Jun Li, Kazutaka Furuse, and Kazunori Yamaguchi. Focused crawling by exploiting anchor text using decision tree. In Allan Ellis and Tatsuya Hagino, editors, *WWW (Special interest tracks and posters)*, pages 1190–1191. ACM, 2005.

[Liu *et al.*, 2004] Hongyu Liu, Evangelos Milios, and Jeannette Janssen. Probabilistic models for focused web crawling. In *WIDM '04: Proceedings of the 6th annual ACM international workshop on Web information and data management*, pages 16–22, New York, NY, USA, 2004. ACM.

[Menczer *et al.*, 2001] Filippo Menczer, Gautam Pant, Padmini Srinivasan, and Miguel E. Ruiz. Evaluating topic-driven web crawlers. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 241–249, New York, NY, USA, 2001. ACM.

[Najork and Wiener, 2001] Marc Najork and Janet L. Wiener. Breadth-first crawling yields high-quality pages. In *Proceedings of the 10th International World Wide Web Conference*, page 114118, Hong Kong, May 2001. Elsevier Science.

[Passerini *et al.*, 2001] Andrea Passerini, Paolo Frasconi, and Giovanni Soda. Evaluation methods for focused crawling. In *Proceedings of the 7th Conference of the Italian Association for Artificial Intelligence, Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2001.

[Pearl, 1985] Judea Pearl. Bayesian networks: A model of self-activated memory for evident ial reasoning. In *Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine*, pages 329–334, 1985.

[Quinlan, 1993] Ross J. Quinlan. *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann, January 1993.

[Quinlan, 1996] J. Quinlan. Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.

[Sahami *et al.*, 1998] Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. A bayesian approach to filtering junk E-mail. In *Learning for Text Categorization: Papers from the 1998 Workshop*, Madison, Wisconsin, 1998. AAAI Technical Report WS-98-05.

[Sizov *et al.*, 2002] Sergej Sizov, Stefan Siersdorfer, Martin Theobald, and Gerhard Weikum. Weikum: The bingo! focused crawler: From bookmarks to archetypes. In *Demo Paper, International Conference on Data Engineering (ICDE*, 2002.

[Umbrich *et al.*, 2008] Jürgen Umbrich, Andreas Harth, Aidan Hogan, and Stefan Decker. Four heuristics to guide structured content crawling. In *Proceedings of the Eighth International Conference on Web Engineering*, 2008.

[Umbrich, 2008] Jürgen Umbrich. Discovering and crawling structured content. Master's thesis, University of Karlsruhe (TH), School of Economics and Business Engineering, 2008.

[URL, a] http://citeseer.ist.psu.edu/.

[URL, b] http://scholar.google.com/.

[URL, c] http://seekda.com/.

[URL, d] http://technorati.com/.

[URL, e] http://tools.ietf.org/html/rfc1436.

[URL, f] http://tools.ietf.org/html/rfc959.

[URL, g] http://www.blinkx.com/.

[URL, h] http://www.iana.org/assignments/mime-types/.

[URL, i] http://www.ietf.org/rfc/rfc2396.txt.

[URL, j] http://www.seeqpod.com/.

Author Index

FGWM 09

Workshop on Knowledge and Experience Management (Wissens- und Erfahrungsmanagement) 2009
Gesellschaft für Informatik, Fachgruppe Wissensmanagement

Editors

Christoph Lange, Jacobs University Bremen
Jochen Reutelshöfer, Universität Würzburg

Workshop on Knowledge and Experience Management 2009

Christoph Lange
Jacobs University Bremen
Germany

Jochen Reutelshöfer
Universität Würzburg
Germany

The FGWM Workshop

The workshop on Knowledge and Experience Management is held as a part of the LWA workshop series and is organized by the Special Interest Group on Knowledge Management (Fachgruppe Wissensmanagement, FGWM – http:/www.fgwm.de) of the German Computer Science Society (Gesellschaft für Informatik, GI).

The Special Interest Group on Knowledge Management addresses automated methods for the capture, the development, the utilization, and the maintenance of knowledge for organizations and networks of people. The main goal of the workshop is the exchange of innovative research ideas and experiences with practical applications in the area of knowledge and experience management.

Thereby, it aims to provide an interdisciplinary forum for researchers and practitioners for exchanging ideas and innovative applications with respect to knowledge and experience management.

FGWM 2009

For the 2009 edition of the FGWM workshop, we accepted 6 full research papers and 5 resubmissions of previously published work. For the latter only one page abstracts are included in the proceedings. Further, 5 posters have been submitted to be presented at the poster sessions of the LWA conference. The topics of interests of the FGWM workshop series are:

- Experience/knowledge search and knowledge integration approaches (case-based reasoning, logic-based approaches, text-based approaches, semantic portals/wikis/blogs, Web 2.0, etc.)
- Applications of knowledge and experience management (corporate memories, e-commerce, design, tutoring/e-learning, e-government, software engineering, robotics, medicine, etc.)
- (Semantic) Web Services for knowledge manangement
- Agile approaches within the knowledge management domain
- Agent-based & Peer-to-Peer knowledge management
- Just-in-time retrieval and just-in-time knowledge capturing
- Ways of knowledge representation (ontologies, similarity, retrieval, adaptive knowledge, etc.)
- Support of authoring and maintenance processes (change management, requirements tracing, (distributed) version control, etc.)
- Evaluation of knowledge management systems
- Practical experiences ("lessons learned") with IT-aided approaches
- Integration of knowledge management and business processes

Program committee

The program committee had the following members (in alphabetical order):

- Klaus-Dieter Althoff, Universität Hildesheim
- Joachim Baumeister, Universität Würzburg
- Ralph Bergmann, Universität Trier
- Ioannis Iglezakis, Universität von Thessaloniki
- Andrea Kohlhase, Jacobs University Bremen
- Mirjam Minor, Universität Trier
- Markus Nick, empolis GmbH
- Ulrich Reimer, University of Applied Sciences St. Gallen
- Thomas Roth-Berghofer, DFKI
- Rainer Schmidt, Universität Rostock
- Steffen Staab, Universität Koblenz-Landau

Program committee

We would like to thank the authors for their submissions, and we also thank the members of the program committee for providing helpful constructive reviews. Our thanks also go to the organizers and supporters of the LWA workshop series, especially Melanie Hartmann and Frederik Janssen.

August, 2009,

Christoph Lange and Jochen Reutelshöfer

Table of Contents

The Usability Stack: Reconsidering Usability Criteria regarding Knowledge-Based Systems

Martina Freiberg, Joachim Baumeister, Frank Puppe
University of Würzburg
D-97074, Würzburg, Germany
freiberg/joba/puppe@informatik.uni-wuerzburg.de

Abstract

Considering usability-based design and evaluation, many general guidelines and heuristics have been proposed in the past. Yet there is still a lack of tailored criteria for the specific case of knowledge-based systems. In this paper we propose the *Usability Stack* for knowledge-based systems as a new model that summarizes and classifies usability criteria for that particular context. The model as a whole, as well as the different layers in particular, are discussed. We also shortly describe the results of exemplarily going through one knowledge system, developed by our department, according to the model.

1 Introduction

Concerning the usability, knowledge-based systems constitute a particular case compared to general software.

Concerning the case of consultation, knowledge-based systems often are not used by their potential users on a regular basis and additionally under considerable strain. This is confirmed by our experience with systems that are applied in the medical domain—examples are systems that support abdomen sonography (SonoConsult, [5]) or dental consultancy (a tailored dialog, used as a case study in [2]). Therefore it is essential that knowledge-based consultation systems are as easily usable and as highly self-descriptive as possible. That way, users can always resume working with the system in an easy and effective way—even after longer or stressful breaks.

Another issue, concerning especially the knowledge acquisition task, is the required high level of expertise concerning both the application domain and knowledge engineering in general. Thus, ideally, knowledge engineers and domain experts collaborate on setting up a knowledge-based system; yet, a cooperation of—a minimum of—two such specialists is often too expensive, for instance see Puppe [13]. In drastically simplifying the usage of knowledge-acquisition tools, also non-experts could more easily support domain specialists in developing knowledge-based systems.

Thus, enhancing the ease of use of knowledge-based systems—regarding both the consultation and knowledge-acqusition use case—could contribute to further extend the range of potential application contexts, and thus increase their distribution. Hence the important, still unmatched, long-term goal is to purposefully enhance their *usability*.

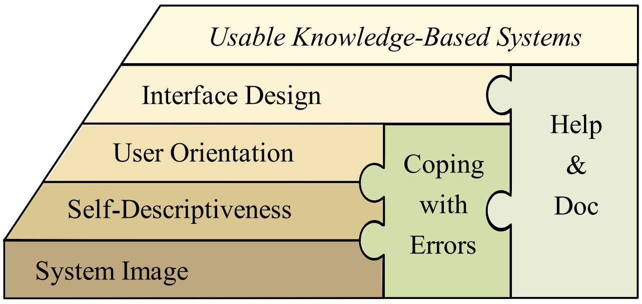


Figure 1: *Usability Stack* for knowledge-based systems

Regarding usability-based design and evaluation, many usability resources have been proposed to date. Yet, concerning knowledge-based systems, those resources often either are lacking specific requirements, or on the other hand partly incorporate insignificant aspects.

In this paper, we propose the *Usability Stack* for knowledge-based systems (depicted in Figure 1)—a model of usability criteria, tailored for knowledge-based systems. By knowledge-based systems we mean systems that utilize problem-solving knowledge and inference mechanisms to support the user in decision-making or in finding solutions for a given problem. At that point in time, we aim at generating a model that is applicable to different classes of knowledge-based systems with respect to their underlying inference method—that is, we made no distinction between, for example, case-based or rule-based knowledge systems so far. Whether and to what extent such a distinction could be additionally useful for our context will be further investigated in the future.

To create a basic stack-model of usability criteria, we reconsider well-known usability resources in detail; as far as necessary, we extend and reassemble usability criteria, which are characterized in section 2. Based on that, we build a stack-model that explains their dependencies and interconnections, described in section 3. Section 4 presents the results of an exemplarily application of the model to a knowledge-based system. We conclude the paper with a short summary of the preliminary results and the future prospects in section 5.

2 Tailored Usability Criteria

In this section we shortly introduce the examined usability resources and give reason for their choice. Afterwards, we present the tailored criteria that we reassembled from the existing resources, and we also explain their relevance for the context of knowledge-based systems.

2.1 Resources

The following usability resources have been taken into account: 10 heuristics of Nielsen [11, Ch. 2], 15 heuristics of Muller et al. [10], 11 heuristics of Constantine & Lockwood [4, pp. 45 ff.], 15 heuristics of Kamper [7], 10 heuristics of Sarodnick & Brau [14, pp. 140 f.], 8 principles of Norman [12], 8 principles of Shneiderman [16, pp. 74 f.], 16 principles of Tognazzini [18], 7 principles of DIN ISO 9241-110 [15], and selected principles from Lidwell's "Universal Principles of Design" [9].

Those resources are widely known and applied for usability evaluation, thus constituting today's state-of-the-art. In addition, there exist many more guidelines or styleguides that describe usability criteria in a more detailed manner—for example, Thissen's *Handbook on Screen Design* [17], or Apple's *Human Interface Guidelines* [1]. Such resources, however, are often already much too detailed, which makes them more difficult to tailor adequately for our purpose. This is the reason why we chose the more general heuristics as an initial point for defining our tailored usability criteria.

2.2 The Usability Layers

We now first describe each layer of the stack—i.e. each usability criterion—separately. Thereby we first name the key requirement of each criterion, and then summarize its different aspects in a list. We further give reasons for their relevance in the context of knowledge-based systems. The composition of the stack as a whole, and the interconnections and dependencies of its layers, are further described in section 3.

System Image

→ ***Key demand:*** *create a tailored and comprehensible system model, according to the system's real-world context*

- apply metaphors, story lines, or visible pictures wherever appropriate
- consider and support users's actual work- and taskflow
- consistency between user expectations and the system
- sequence, structure, and group tasks in a natural/logical way
- break down complex tasks into smaller, more intelligible subtasks
- leave simple tasks also simple within the system

We regard the application of an appropriate system image as highly relevant for knowledge-based systems. Due to the often necessary specific expertise, it is all the more important to facilitate the general usage of the system. Here, a tailored metaphor can considerably ease comprehending the system and its mode of operation. Yet, its tailoring to the real-world context and users is essential as to ensure that it is correctly understood; thereby, also user expectations towards the system should be taken into account. As knowledge-based systems often require complex procedures according to their application domain, simplifying tasks—in breaking them down into more intelligible subtasks—can radically ease the system use and thus enhance its usability. A natural and logical task ordering/grouping according to the real-world context should be provided to further increase the users' understanding of the system; this also includes the demand to leave tasks, that are simple by their nature, also simple within a system.

One notably positive example for a well-implemented system image is the consultation and documentation system for dental findings (used as a case study in Atzmueller et al. [2]). As Figure 2 shows, the model of a tooth diagnosis sheet from the daily dental work context was transferred into the web interface in adopting the numbering and alignment of quadrants (Figure 2, I-IV) and teeth (Figure 2, 11-48). Regarding users from the dental medical domain, such a model eases the system's usage drastically.

Self-Descriptiveness

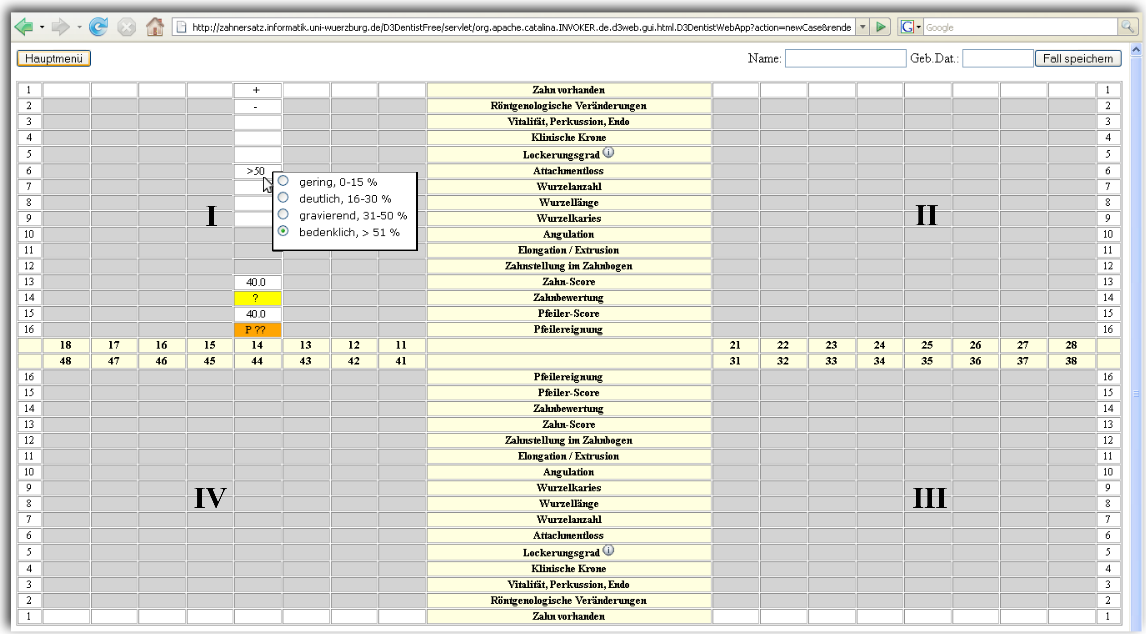
→ ***Key demand:*** *ensure transparent actions and meaningful feedback*

- make potential actions and options visible and accessible (transparent)
- always provide feedback within reasonable time
- illustrate which actions caused the current system state, and which actions are suggested next
- messages: clear, unambiguous, and concise; preferably in plain language
- in the case of errors, clearly communicate the problem and suggest steps for recovery
- also use visual means for communication/illustration

Regarding the use case of knowledge-based systems as consultation systems, immediate feedback on the current system status seems to be especially valuable; communicating also minor, intermediate results (solutions), that occur during the consultation process, allows the user to benefit at any time from using the system—even in the case of system errors or quitting the system ahead of time. In the consultation context it is also helpful to illustrate how the current state (the currently derived solution) was achieved, and what actions (questions to answer) are suggested next. Furthermore, potential actions and options should be visible and accessible within the system.

An example, where this aspect has been implemented well, is the d3web.Dialog2 system (Krawczyk [8], shown in Figure 3). This consultation system, based on a question and answer metaphor, makes use of the AJAX-technology to provide immediate feedback in displaying also intermediate results (Figure 3, a) as soon as the user provides input—also, already answered questions are greyed out instantly (Figure 3, b), thus illustrating, which answers have led to the currently derived solution(s); furthermore, the next suggested questions are highlighted through a yellow background (Figure 3, c), and potentially additional follow-up questions are presented at once when necessary.

With respect to the data acquisition use case, making all required actions available should be—as for other kinds of software, too—a matter of course. Additionally highlighting the next suggested actions or input offers a high potential to ease the system usage and thus enhance overall usability. Another key aspect in our context is the appropriate communication prior to or in the case of actually occuring errors; its relevance for knowledge-based systems is discussed in the corresponding section *Coping with Errors* afterwards. Finally, we regard visual means an important requirement to enhance the self-desciptiveness of a knowledge-based system—visualizations are able to present even complex correlations (e.g. the structure of a knowledge base, or navigation support in a consultation system) in a condensed and illustrative way, which in turn improves the users' understanding of the system, and thereby contributes to an enhanced usability.



http://zahnersatz.informatik.uni-wuerzburg.de/D3DentistFree/servlet/org.apache.catalina.INVOKER.de.d3web.gui.html.D3DentistWebApp?action=newCase&rende
Google
Hauptmenü
Fall speichern
I
II
III
IV

Name: Geb.Dat.:

1					+				Zahn vorhanden									1
2					-				Röntgenologische Veränderungen									2
3									Vitalität, Perkussion, Endo									3
4									Klinische Krone									4
5									Lockerungsgrad ⓘ									5
6					>50				Attachmentloss									6
7									Wurzelanzahl									7
8									Wurzellänge									8
9									Wurzelkaries									9
10									Angulation									10
11									Elongation / Extrusion									11
12									Zahnstellung im Zahnbogen									12
13					40.0				Zahn-Score									13
14					?				Zahnbewertung									14
15					40.0				Pfeiler-Score									15
16					P ??				Pfeilereignung									16
	18	17	16	15	14	13	12	11		21	22	23	24	25	26	27	28	
	48	47	46	45	44	43	42	41		31	32	33	34	35	36	37	38	
16									Pfeilereignung									16
15									Pfeiler-Score									15
14									Zahnbewertung									14
13									Zahn-Score									13
12									Zahnstellung im Zahnbogen									12
11									Elongation / Extrusion									11
10									Angulation									10
9									Wurzelkaries									9
8									Wurzellänge									8
7									Wurzelanzahl									7
6									Attachmentloss									6
5									Lockerungsgrad ⓘ									5
4									Klinische Krone									4
3									Vitalität, Perkussion, Endo									3
2									Röntgenologische Veränderungen									2
1									Zahn vorhanden									1

○ gering, 0-15 %
○ deutlich, 16-30 %
○ gravierend, 31-50 %
● bedenklich, > 51 %

Figure 2: Positive Example for a Tailored System Metaphor, from a Dental Documentation/Consultation System

User Orientation

→ ***Key demand:*** *implement reasonably constrained user control in a system that is tailored for its potential users*

- provide control through undo & redo, pause & resume
- let users sequence tasks according to their needs
- avoid surprising/awkward system behavior
- deactivate or hide dangerous/critical actions
- constrain critical/complex input
- account for target users' skills, demands, and habits
- speak the users' language

Especially with respect to the often complex tasks occuring in our context, we regard the implementation of constraints, to avoid potential system errors or defective input, as extremely valuable. Critical or dangerous options/actions should be deactivated or completely hidden in the interface. Also pausing and resuming options, as well as the possibility to carry out tasks in any given order, enhance the usability of a knowledge-based system.

Again, the dental documentation and consultation system provides for a good example of reasonably constrained user control. Figure 2 shows that input fields for a tooth are not editable until certain input is provided; this slight constraint assures reasonable input based on previous actions—yet, the user controls in what order to provide the input, once the fields are editable; moreover its the user who generally decides about the order or number of teeth to edit, thus having a great deal of control of this system. The d3web.Dialog2 system (Figure 3), in contrast, provides user guidance through the dialog to a much greater extent—in suggesting the appropriate order of questions, and also in providing follow-up questions straightly dependent on previous input. Yet, in the end the user remains in control also of this dialog, as it is also possible to leave the suggested path and chose a different order of answering the questions.

Apart from a reasonably constrained user control, we also regard tailoring a system to its potential users as highly relevant. Characteristically, knowledge-based systems are applied in quite specific contexts—thus, for example, medical professional terminology is first-choice for medical staff, whereas this would not be appropriate for biologists. Also, differing user groups often possess different skills and preferred taskflows (e.g. physicians vs. nurses). Tailoring a knowledge-based system to its potential users thus can strongly contribute to its understandability, and consequently to its usability and user efficiency.

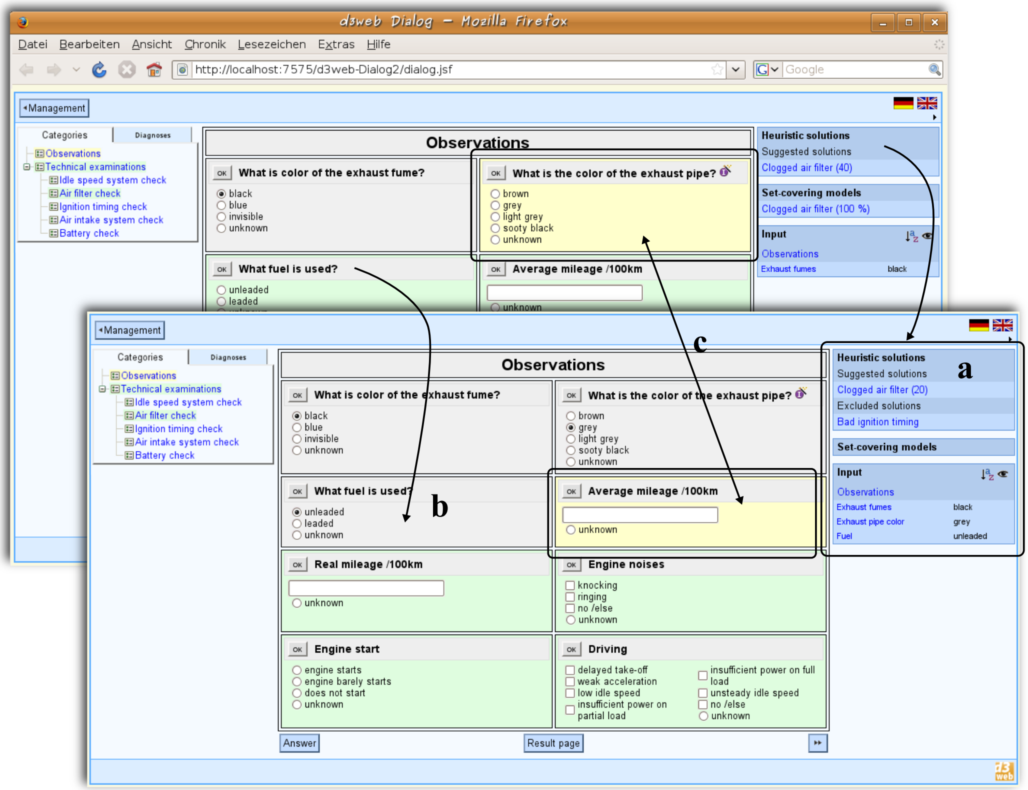
Interface Design

→ ***Key demand:*** *design an aesthetical, yet minimalistic interface*

- hide actions and options not straightly needed from the interface (minimalistic interface)
- clear conveyance and presentation of information, highlighting of important information
- no multi-page displays
- design- and behavior-based consistency
- consistency with standards/conventions from application context
- consider general design guidelines (colors etc.)
- aesthetical, pleasing, "modern" design
- provide distinct "views", where appropriate

Again due to the often complex application domains and procedures related with knowledge-based systems, both the minimalistic and consistent design of an interface are of high priority. Thereby, the former induces simplicity, as important information and required functionalities can be found more easily; the latter, consistency, avoids confusion on the side of the user, which allows for concentration on the task at hand and thus also eases the system usage.

Furthermore, software interfaces should in general be designed in an aesthetic, pleasing way, concerning, for example, the adherence to general design-based principles regarding the choice of colors, fonts, alignment issues etc.



Observations

What is color of the exhaust fume?
● black
○ blue
○ invisible
○ unknown

What is the color of the exhaust pipe?
○ brown
○ grey
○ light grey
○ sooty black
○ unknown

What fuel is used?
○ unleaded
○ leaded

Average mileage /100km
○ unknown

Heuristic solutions
Suggested solutions
Clogged air filter (40)

Set-covering models
Clogged air filter (100 %)

Input
Observations
Exhaust fumes black

Observations

What is color of the exhaust fume?
● black
○ blue
○ invisible
○ unknown

What is the color of the exhaust pipe?
○ brown
● grey
○ light grey
○ sooty black
○ unknown

What fuel is used?
● unleaded
○ leaded
○ unknown

Average mileage /100km
○ unknown

Real mileage /100km
○ unknown

Engine noises
□ knocking
□ ringing
□ no /else
○ unknown

Engine start
○ engine starts
○ engine barely starts
○ does not start
○ unknown

Driving
□ delayed take-off
□ weak acceleration
□ low idle speed
□ insufficient power on partial load
□ insufficient power on full load
□ unsteady idle speed
□ no /else
○ unknown

Heuristic solutions
Suggested solutions
Clogged air filter (20)
Excluded solutions
Bad ignition timing

Set-covering models

Input
Observations
Exhaust fumes black
Exhaust pipe color grey
Fuel unleaded

Figure 3: Positive Example for Immediate Feedback, Implemented in a System for Car Fault Diagnosis

This may not directly influence usability, yet it can enhance the user's perception of the system; in turn, a system that is perceived as pleasing is often assumed to be more usable than other systems. In this regard, an interface should as well appear somewhat "up to date"; such interfaces are more likely presumed to be well maintained and tailored for today's hardware. Thus they consolidate the user's confidence towards the system. Concerning the context of knowledge-based systems, we regard the aesthetical aspect also as relevant as it can—not solely but additionally—enhance the user's experience of the system.

Knowledge-based systems often address users from several considerably distinct user-groups—in the case, that those can be clearly specified, we regard the provision of tailored views of the system as highly valuable. Such a view, however, should not only refer to the outer appearance, as e.g. color scheme or alignment. Rather the system as a whole should be affected, thereby considering the specific skills of each user-group. This includes, for example, structuring tasks differently or providing for other kinds of functionalities (e.g. wizard-guided vs. self-guided interfaces).

Coping with Errors

→ ***Key demand:*** *anticipate and prevent errors whenever possible, in the case of errors provide help for recovery*

- apply appropriate error prevention measures
- hide or deactivate potentially critical/dangerous options and actions
- provide undo & redo for leaving critical states
- concise, comprehensible, informative error messages
- suggest steps for recovery
- handle error internally until problem can be resolved
- prevent data loss

Regarding knowledge-based systems, this aspect also is highly relevant. Those systems generally require a lot of data input, independent of whether they are applied for data acquisition (e.g. setting up the knowledge base) or for consulting (e.g. gathering information to derive solutions)—thus a sudden data loss due to system errors is extremely painful for the users. That case should be prevented through adequate and anticipatory communication (warnings, suggestions of alternative actions or options) and constraints (hiding and deactivation of critical actions and options). Additionally, errors should be prevented through undo & redo facilities, default value presentation, auto-completion, or the intelligent suggestion or highlighting of appropriate next actions. Moreover, also consistency tests should be implemented both on a global and a local basis; examples are limiting the presented questions in a consultation system to the reasonable follow-up questions (global), and verifying critical input to the reasonably processable options (local). If specific critical system states are completely unavoidable, at least tailored, comprehensible error messages should be provided; thereby not only the type and context oft the error should be explained, but adequate steps for recovery should be suggested. At any rate, errors should be handled internally without resulting in data loss, or an awkward or surprising system behavior, thus maintaining a stable-running system.

Help & Doc

→ ***Key demand:*** *provide easily accessible and tailored help and documentation*

- easily accessible at any time and system state
- only as large and detailed as necessary
- searchable index for easier information retrieval

- tailor help & doc to actual system purpose, thus refer to and support actual tasks
- where appropriate, provide tutorials, step-by-step instructions, examples, or reference of the system

It is often stated, that help and documentation are not so important as most people never use them anyway—nevertheless we regard that criterion as a relevant, additional means for improving overall usability in the context of knowledge-based systems. Considering the often tedious tasks and procedures—which cannot always be simplified—an accessible and supportive help- and doc system can be of great value, especially for novice users. Thereby an introductory listing and explanation of best practices or standard tasks regarding the system use can help such users get started. Also a searchable, indexed reference system that covers potential actions and options, specific forms of data input, or that explains more complex parts of the interface can add value to the system. We generally agree, that a system should best be usable without any help at all, yet we consider the specific context as often far too complex to actually completely fulfil this demand; thus, a tailored help & doc system can improve the user's system experience—as it ideally enables the user to solve tasks and even problems on his own—and with that the perceived usability.

3 Composing the Usability Stack

In the previous section 2 we described the different components of the *Usability Stack*. Now we clarify its composition and explain the interconnections and dependencies of the layers. Figure 1 depicts the stack of usability criteria. We begin by explaining the four layers on the lefthand side:

- System Image
- Self-Descriptiveness
- User Orientation
- Interface Design

Layer 1—System Image

We chose *System Image* as the basis of the model because it strongly influences all the other criteria that build upon it; also, it is interdependent with the two additional criteria, depicted as the vertical layers, which will be illustrated in more detail in the subsequent sections. Basically, we not only belief that a tailored model or metaphor can greatly enhance a system's usability, but also, when inappropriately chosen (for the given context or target user group), can turn a system nearly worthless. Moreover we consider the real-world context of a system—as taken into account also within the first layer—an essential basis for the following criteria. Thus we regard the development of an adequate system image as the most important, foundational task when striving for a usable knowledge-based system.

Layer 2—Self-Descriptiveness

An appropriate system image then already contributes greatly to *Self-Descriptiveness*; the former enables users to better predict the system's behaviour, which in turn largely meets the request, that a system should guide users by its design and ability to clearly convey necessary information. The chosen system model or metaphor also influences the appropriate affordances and hints that are to be provided—as useful hints for one context are likely to be worthless for another context; likewise also necessary actions and options vary depending on the application context or the appropriate task structure. So in summary, self-descriptiveness directly depends on the realization of the system image, which is why we regard it as the second most important criterion for knowledge-based systems and thus as the logical second layer of our model.

Layer 3—User Orientation

User Orientation forms the subsequent third layer within our model. Tailoring a system for its intended target users is highly dependent on the real-world context (as considered within the first layer) that already defines the target users and their specific wants and skills. Also developing and implementing reasonably constricted user control depends on the previous layers. On the one hand, the chosen system image (first layer) often dictates which parts of a system should be controlled by the users and to what extent; on the other hand, this criterion also builds on the findings, which actions and options should be visible at what system state (outcome of the second layer), as only actions/options that are actually visible can be discussed to be constrained or fully controllable at all.

Layer 4—Interface Design

As the three previous layers all influence the final form of the interface, the *Interface Design* layer is stacked on top of the others. First, the chosen system image (first layer) often specifies quite detailed what an appropriate interface looks like as a whole; this is also dependent on the grouping and structuring of tasks. Influencing findings from the second layer are the necessay dialogs and messages required, necessary progress indicators, or the potential need for providing visual forms of navigation support, system overviews, or other kinds of information visualizations. Finally, the analysis of target user groups and their specific wants and needs from the third layer can also affect the interface design: in prescribing, which actions and options should be deactivated for constraining reasons, or if distinct user groups require specific views of the system.

Examining Figure 1 again, however, the *Interface Design* layer builds not only on the lefthand side stacked layers, but also on the vertical layer *Coping with Errors*. This is because we regard it important to first have set up a stable system before thinking about how to present that system to the user. In our opinion, it has first—in the context of the criterion *Coping with Errors*—to be decided, which actions to take to assure the best possible error handling. Based on decisions, as where to display which kind of confirmation or warning/error messages, where to present default values, or where to even hide parts of the interface, an appropriate interface design with tailored dialogs can be created.

In addition to the four layers on the lefthand side, the stack also consists of two criteria that do not built on just exactly one of the other critera but are interrelated with several ones:

- Coping with Errors
- Help and Doc

In the overview in Figure1, those two criteria are depicted as vertical layers, interconnected with several other layers of the stack. In the following, they are described in more detail.

Layer 5—Coping with Errors

To begin with *Coping with Errors*, we already pointed out above why this layer constitutes an additional foundation for the fourth stacked layer, interface design. Thus it remains to explain in what ways *Coping with Errors* is dependent or interconnected with the three remaining layers on the left. To begin with the first layer, it is coupled with the fifth criterion as the system model should be designed in a way that it is easily comprehensible by the users and thus reduces potential errors; this also includes the way, how tasks should be broken down (if necessary) and structured. Next, there exists a connection to the second stacked layer, as an important aspect of usable systems is to clearly communicate occuring problems and suggest solutions in the case of errors. Also the demand for suggesting/highlighting the next appropriate actions relates to *Coping with Errors*, as the appropriate implementation also can contribute to error prevention. The third stacked layer finally relates to *Coping with Errors* with respect to the decisions which actions to constrain and to what extent, to prevent severe system errors.

Layer 6—Help & Doc

Help and Documentation finally is a criterion, that in one way or another is related to all of the previously described criteria; thus it is depicted as a vertical layer connected with all other layers on the left. To begin with the system model, yet also the user interface, those should be introduced and explained in some form of documentation; thereby the system's general mode of operation (defined by the system image, first layer) as well as how to conduct specific tasks by pointing out which parts of the interface make available the necessary actions and options (defined by the final interface design, fourth layer) should be explained. Also, easily accessible help—say, in form of a context sensitive help system, or automatically suggesting appropriate actions or problem solutions—can enhance the self-descriptiveness of a system. With regards to the third stacked layer, user orientation, as well as to the criterion coping with errors, a help system documenting existing constraints within the system is valuable in case a user starts wondering why certain actions or options are not always available. Also, help & documentation ideally enable users to solve tasks or even errors on their own, thereby providing them a great sense of mastery of the system.

The final layer, *Usable Knowledge-Based Systems*, building on all the other ones, finally depicts the belief that usable knowledge-based systems can be the outcome of designing—or evaluating and adapting—systems according the *Usability Stack*.

4 Applying the Usability Stack to the Semantic Wiki KnowWE

This section presents the results of exemplarily applying the stack-model to one of the knowledge-based systems recently developed by the Department of Artificial Intelligence and Applied Informatics, University of Würzburg. For this purpose we chose KnowWE, a wiki-based system for collaborative knowledge engineering and knowledge-based consultation (Baumeister et al. [3]). Figure 5 shows the car diagnosis knowledge-base—supporting users in detecting car-related malfunctions—as represented in KnowWE.

4.1 System Image

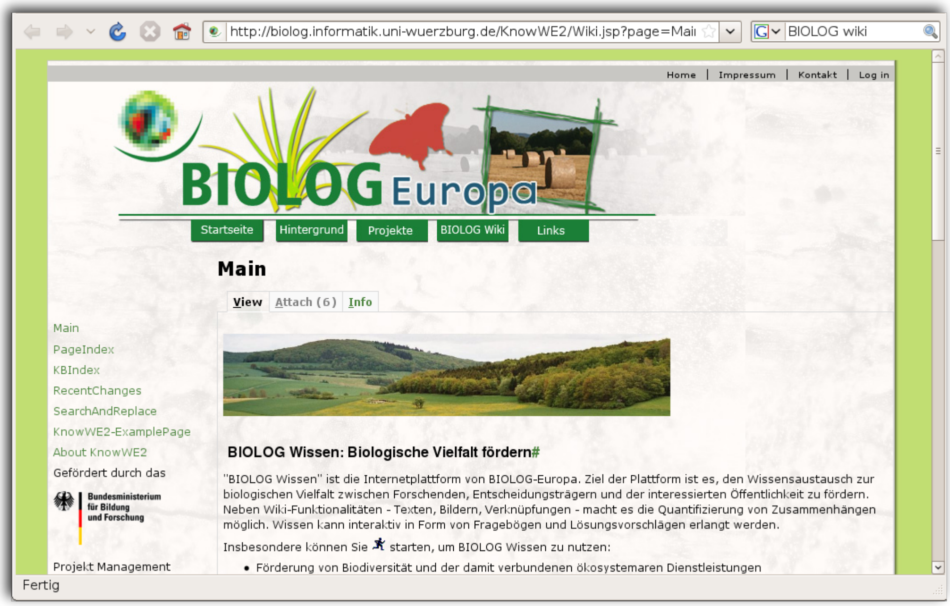
The system image, provided by KnowWE, is the standard wiki model. Wiki systems are widely spread today and thus their general mode of operation should be known to most people using a computer regularly. This eases getting started with KnowWE at least concerning the standard tasks for potential users of the system, as such tasks—for example, creating a wiki-page—basically conform to the standard wiki way. The use of the built-in consultation features as, for example, the embedded dialog or the inplace-answer lists, are quite intuitive: in the case of the dialog, the user selects the appropriate answers by just clicking on them (Figure 6, a), and in the case of the question sheet, a click on the question category opens a small pop-up containing the answer possibilities (Figure 6, b)—which again can be selected by a click. Once a user starts answering the questions, solutions are derived after each input and displayed in the solutions panel (Figure 5, b). Yet to take advantage of KnowWE's data acquisition features, the particular knowledge markup has to be learned additionally. As KnowWE aims at supporting different forms of knowledge, the markup language has already grown quite complex, and thus probably poses some problems for novice users, or such users that cannot use the system regularly. Some experiments with KnowWE demonstrated that the knowledge-markup, although it is quite easily learnable, still leaves room for potential mistakes.

4.2 Self-Descriptiveness

The second criterion, self-descriptiveness, could so far achieved in parts. First, mostly appropriate feedback concerning the system status is communicated. When using KnowWE for knowledge acquisition—entering and modifying knowledge in a wiki article (Figure 5, a)—the results are immediately visible either when the article is saved or via a preview-feature (Figure 5, d). Concerning the use of KnowWE as consultation system, the recently derived solutions (Figure 5, b) are displayed and updated immediately, according to the latest user input; thus, also intermediate results are accessible. Yet, self-descriptiveness of the system as a whole has to be further improved—basically, the offered feedback is adequate for users that know how to achieve certain goals. Regarding novice users, though, more feedback to get them started—for example, suggesting first actions, or better highlighting the general tasks that can be accomplished—should be provided. Concerning potential errors, in some critical cases warnings are provided—one example is, that a user has to confirm a warning before saving an article, if that page is at the same time being edited by another user. Additional future improvements of self-descriptiveness could be achieved by visualizing the navigation or an overview of the system and its linkages to further enhance the understandability of the system as a whole and thus its usability.

4.3 User Orientation

User orientation is a two-sided issue within KnowWE. On the one hand, the chosen model of a wiki provides for a reasonable user control—the user can decide in person whether to use existing knowledge by just browsing through the pages (consultation), or when and to what extent to enter or modify knowledge (knowledge acquisition). Those two main tasks can also be cancelled at any time ("emergency exits“). To improve this criterion even more, pause and resume actions regarding the data acquisition



BIOLOG Wissen: Biologische Vielfalt fördern#

"BIOLOG Wissen" ist die Internetplattform von BIOLOG-Europa. Ziel der Plattform ist es, den Wissensaustausch zur biologischen Vielfalt zwischen Forschenden, Entscheidungsträgern und der interessierten Öffentlichkeit zu fördern. Neben Wiki-Funktionalitäten - Texten, Bildern, Verknüpfungen - macht es die Quantifizierung von Zusammenhängen möglich. Wissen kann interaktiv in Form von Fragebögen und Lösungsvorschlägen erlangt werden.

Insbesondere können Sie starten, um BIOLOG Wissen zu nutzen:

- Förderung von Biodiversität und der damit verbundenen ökosystemaren Dienstleistungen

Figure 4: BIOLOG Wiki (a Collaboration System for Researchers on Biological Diversity)—Based on KnowWE

task should be considered; this would enable a user not only to completely cancel data entry, but also to resume it from exactly the point he left before and thus not having to start from the beginning. Undo and redo options are currently realized through a version control mechanism that enables users to restore a previous version of the entire page. Using undo while editing a page, though, is not yet implemented so the user has to remember his last edits to be able to undo them manually; once having left the editing mode of a page, undoing the last edits is also not yet possible; still such undo features would be preferable to ease retracting minor mistakes.

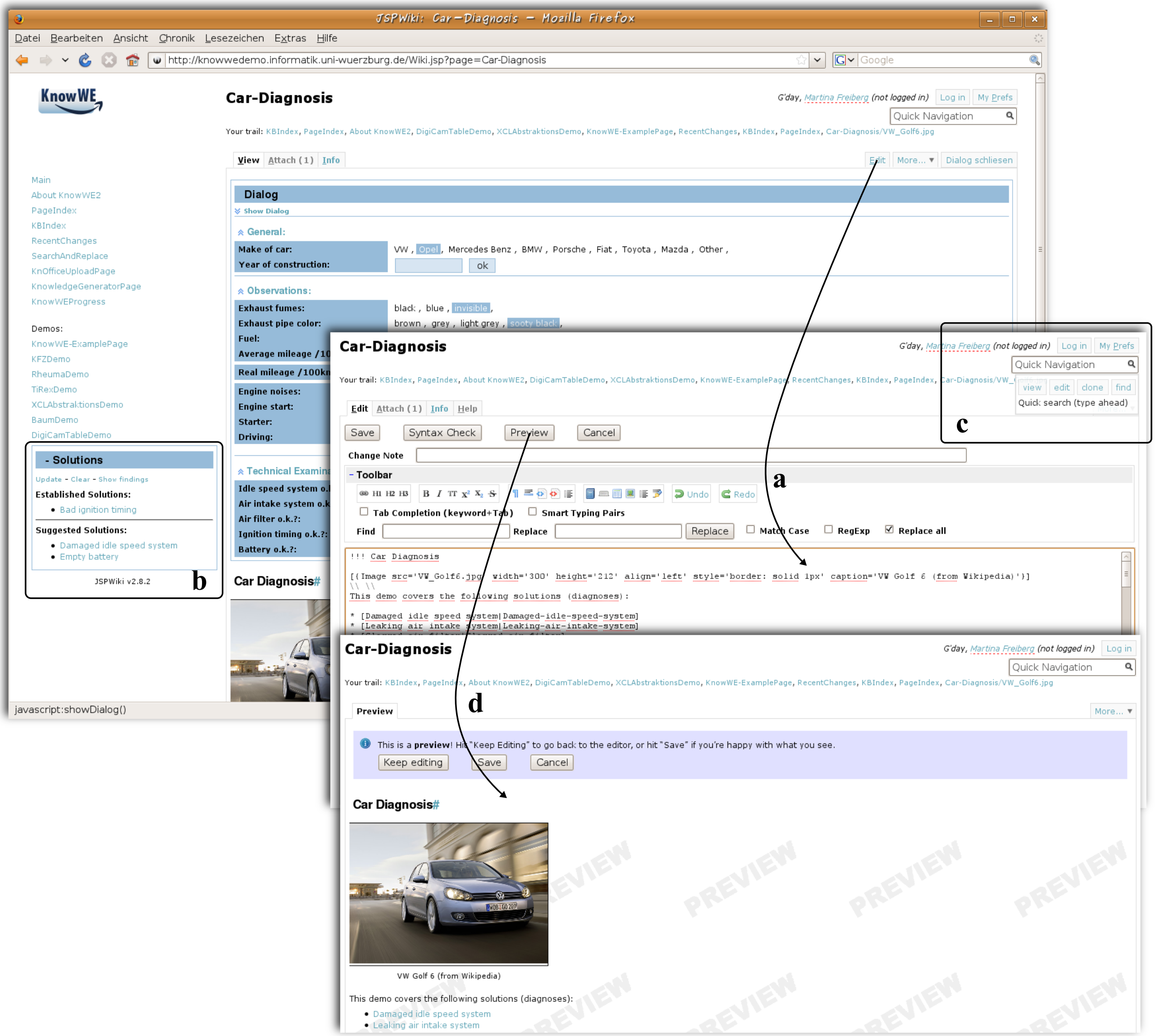
The second main aspect of user orientation, however—tailoring the system to particular user groups—is not yet implemented in the basic KnowWE system. The system has so far been designed on a more general level with the goal to be applicable in several contexts and for different user groups. With certain effort, however, it is possible to achieve tailoring to some extent, as the example of the BIOLOG-wiki (see Figure 4)—that is built based on KnowWE—shows. Yet, some tailoring options—for example several kinds of consultation dialogs, or different editor styles—should be included in the basic system as well to enhance user orientation already there, and further ease future tailoring to distinct user groups.

4.4 Interface Design

Interface Design is one aspect of KnowWE that definitely has more attention to be paid to in the future. Although a quite consistent color scheme and appropriate fonts have been applied, the overall aesthetical presentation of KnowWE still needs improvement. Moreover, some parts of the interface should be enhanced with regards to a minimalistic design. The upper right part of the wiki is such an example—in the case that the history of already visited wiki sites contains various objects, the popup of the search-input field partly overlaps the site history and thus contributes to a quite cluttered impression of this part of the interface (Figure 5, c). Another example is the presentation of the actions and options while editing a wiki page; here, the user is offered the whole palette of editing options and additional features all at once and in too many different ways (buttons, checkboxes, tabbed panes). Another issue with the design of KnowWE is consistency. First, sometimes actions and options are not yet displayed in a consistent way. As Figure 6 depicts, the answer popup for question *fitness* contains an answer-unknown option (represented by *-?-*)—within the dialog, this option is not given, though. Moreover, as KnowWE is a browser-based application, its appearance could slightly differ from one operating system to anothe, as as well within different browsers. This has to be taken special care of in the future, when further refining the interface. Implemented quite well already, however, is the highlighting of important information—recently derived solutions—in KnowWE by displaying them in a distinct labeled box underneath the navigation menu (Figure 5, c).

4.5 Coping with errors

KnowWE in parts already provides for an anticipation of errors in presenting warnings in the case of critical actions. As already mentioned, if a user tries to save a page that is being edited by a second user at the same time, a warning is displayed to avoid the potentially occuring problems (data loss for the other user, or system problems due to saving the same page at the same time). Yet the user still remains in control—that is, if the user wants to, page saving can be continued after confirming the warning. For improving data safety in general, some kind of sophisticated automatic saving mechanism that takes into account the recent edits of both users could be considered. Another aspect implemented regarding error prevention is an auto-completion mechanism when entering knowledge into the wiki. In the case that, regardless of the auto-completion suggestions, some input has been entered inaccurately, an automatic display of suggestions for the correct input format, or at least an informing message, could be additionally valuable. If an error occurs, at least there exists the possibility of restoring the latest, successfully stored version of the page via the provided version control.



```
!!! Car Diagnosis

[{Image src='VW_Golf6.jpg' width='300' height='212' align='left' style='border: solid 1px' caption='VW Golf 6 (from Wikipedia)'}]
\\ \\
This demo covers the following solutions (diagnoses):

* [Damaged idle speed system|Damaged-idle-speed-system]
* [Leaking air intake system|Leaking-air-intake-system]
```

This is a **preview**! Hit "Keep Editing" to go back to the editor, or hit "Save" if you're happy with what you see.

Car Diagnosis#

This demo covers the following solutions (diagnoses):

- Damaged idle speed system
- Leaking air intake system

Figure 5: The Knowledge Wiki Environment KnowWE

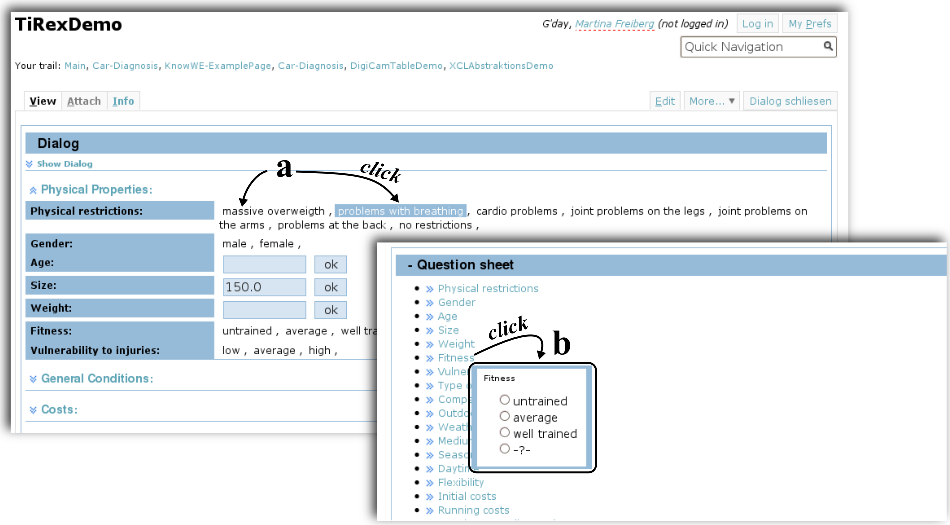

TiRexDemo
G'day, Martina Freiberg (not logged in) Log in My Prefs
Quick Navigation
Your trail: Main, Car-Diagnosis, KnowWE-ExamplePage, Car-Diagnosis, DigiCamTableDemo, XCLAbstraktionsDemo
View Attach Info
Edit More... Dialog schliesen
Dialog
Show Dialog
a
click
Physical Properties:
Physical restrictions:
massive overweigth, problems with breathing, cardio problems, joint problems on the legs, joint problems on the arms, problems at the back, no restrictions,
Gender:
male, female,
Age:
ok
Size:
150.0
ok
Weight:
ok
Fitness:
untrained, average, well tra
Vulnerability to injuries:
low, average, high,
General Conditions:
Costs:
- Question sheet
Physical restrictions
Gender
Age
Size
Weight
Fitness
Vulne
Type
Comp
Outdo
Weat
Mediu
Seaso
Dayti
Flexibility
Initial costs
Running costs
click
b
Fitness
○ untrained
○ average
○ well trained
○ -?-

Figure 6: Consultation Features of KnowWE

4.6 Help and Documentation

KnowWE already provides some exemplary documentation; there exists an introductory page that explains how to use the specific kinds of knowledge markup with the help of actual examples. Yet, this existing documentation should be further extended. Potential examples are step-by-step listings of default tasks or of an introductory tutorial, to get the users started initially, or a searchable index of help and documentation topics that also contains links to the actual support sites.

5 Conclusion

In this paper we discussed the need for a more tailored set of usability criteria regarding the design and evaluation of knowledge-based systems. We proposed a novel model—the *Usability Stack* for knowledge-based systems—for summarizing and classifying usability criteria that are relevant in the context of knowledge-based systems.

Some preliminary results concerning the usability evaluation of knowledge-based systems could be gained from analyzing several distinct systems according to a set of usability criteria extracted from known, usability-related literature (described in [6]). This mainly showed the need of creating a new model of tailored usability criteria for knowledge-based systems. Based on those experiences, we reassembled relevant criteria and worked out the model of the *Usability Stack*. Applying the *Usability Stack* to a recently developed knowledge-system, KnowWE, showed the general applicability of the model. Yet it also revealed the need to refine and distinguish aspects of the model with respect to the two main differing use cases of knowledge-based systems, knowledge engineering and consultation. Concerning the particular inspected system, KnowWE, it turned out that some aspects of the usability criteria have been implemented intuitively in the past. Yet, many other aspects that still offer room for improvement could be identified. Thus future work will include the improvement of the KnowWE system by means of the insights gained from this first examination. Thereby, all of the developed criteria will have to be reconsidered; yet special emphasis will be put on self-descriptiveness, user orientation, and interface design, as those have the most deficiencies so far.

Mainly, however, the model will be applied for the evaluation and redesign of other knowledge-based systems; through an ongoing process of applying and refining the model—thereby working out characteristics of the two main use cases knowledge engineering and consultation—we hope to further improve the model, resulting in a detailed catalog of usability criteria and a related process model of their application. Also we hope to gain insights, whether the model should rather be kept generally aplicable for different classes of knowledge-based systems with respect to the underlying inference methods, or whether a distinction should be made for those cases.

Finally, we also intent to develop and offer some tool to support and ease the application of the stack-model. Such a tool should provide the elaborated criteria in an easy-to-adapt and easy-to-reuse electronic version—for example, in the form of an electronic checklist. Further it would be interesting to provide such a tool with the ability to preliminarily assess and evaluate a knowledge-based system's level of usability, and provide hints to system developers as to which parts of the system require the most refinements with regards to an overall usability enhancement.

References

[1] Apple Computer Inc., Apple Human Interface Guidelines: The Apple Desktop Interface, Addison-Wesley, 1987.

[2] M. Atzmueller, J. Baumeister, A. Hemsing, E.-J. Richter, F. Puppe, Subgroup Mining for Interactive Knowledge Refinement, in: AIME'05: Proceedings of the 10th Conference on Artificial Intelligence in Medicine, Springer, 2005, pp. 453–462.

[3] J. Baumeister, J. Reutelshoefer, F. Puppe, KnowWE: Community-Based Knowledge Capture with Knowledge Wikis, in: K-CAP '07: Proceedings of the 4th international conference on Knowledge capture, 2007.

[4] L. L. Constantine, L. A. D. Lockwood, Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design, Addison-Wesley Professional, 1999.

[5] Frank Puppe and Martin Atzmueller and Georg Buscher and Matthias Huettig and Hardi Lührs and Hans-Peter Buscher, Application and Evaluation of a Medical Knowledge-System in Sonography (SonoConsult), in: Proc. 18th European Conference on Artificial Intelligence (ECAI 20008), accepted, 2008.

[6] M. Freiberg, Usability assessment of knowledge-based consultation systems, Tech. Rep. 457, Institute of Computer Science, University of Würzburg (2009).

[7] R. J. Kamper, Extending the Usability of Heuristics for Design and Evaluation: Lead, Follow, and Get Out of the Way, International Journal of Human-Computer Interaction 14 (3&4) (2002) 447–462.

[8] A. Krawczyk, Konfigurierbarer Dialog zur Nutzung von wissensbasierten Systemen, Master's thesis, University of Würzburg, Germany (2007).

[9] W. Lidwell, K. Holden, J. Butler, Universal Principles of Design, Rockport Publishers Inc., 2003.

[10] M. J. Muller, L. Matheson, C. Page, R. Gallup, Participatory Heuristic Evaluation, Interactions 5 (5) (1998) 13–18.

[11] J. Nielsen, Heuristic Evaluation, John Wiley & Sons, Inc, 1994, pp. 25–62.

[12] D. A. Norman, The Design of Everyday Things, The MIT Press, 1988.

[13] F. Puppe, Knowledge reuse among diagnostic problem solving methods in the shell-kit d3, International Journal of Human-Computer Studies 49 (1998) 627–649.

[14] F. Sarodnick, H. Brau, Methoden der Usability Evaluation: Wissenschaftliche Grundlagen und praktische Anwendung [Usability Evaluation Techniques—Scientific Fundamentals and Practical Applicability], Huber, Bern, 2006.

[15] W. Schneider, Ergonomische Gestaltung von Benutzungsschnittstellen: Kommentar zur Grundsatznorm DIN EN ISO 9241-110, Beuth Verlag, 2008.

[16] B. Shneiderman, C. Plaisant, Designing the User Interface: Strategies for Effective Human-Computer Interaction, Addison Wesley, 2004.

[17] F. Thissen, Screen-Design Handbuch [Handbook on Screen Design], Springer-Verlag, 2001.

[18] B. Tognazzini, The First Principles of Interaction Design, retrieved June 25, 2008 from http://www.asktog.com/basics/firstPrinciples.html.

Modelling Diagnostic Flows in Wikis (Position Paper)

Reinhard Hatko[1], Volker Belli[2], Joachim Baumeister[1]
[1]University of Würzburg, Würzburg, Germany
{hatko, baumeister}@informatik.uni-wuerzburg.de
[2]denkbar/cc, Gerbrunn, Germany
volker.belli@denkbar.cc

Abstract

The engineering of diagnostic process knowledge is a complex and time-consuming task. Diagnostic flows intuitively represent diagnostic problem-solving knowledge by using the workflow metaphor, and they can be interpreted as an generalization of decision trees. In this position paper, we propose a graph-based language for the definition of diagnostic flows and informally describe the basic elements of the language. Due to the modularity of diagnostic flows the collaborative acquisition of the knowledge becomes possible. In the past, Semantic Wikis showed their suitability for the collaborative development of knowledge bases. We introduce a prototypical flow editor, that was implemented as a plugin of the Semantic Wiki KnowWE.

1 Introduction

In general, knowledge can be classified into declarative and procedural knowledge. While the declarative part of the knowledge encompasses the facts and their relationships, the procedural one reflects the knowledge about how to perform some task, i.e. deciding which action to take next. In diagnostics the declarative knowledge particularly consists of findings, diagnoses and sometimes also therapies and their interrelation. The procedural knowledge for diagnosis in a given domain is responsible for the decision which examination to perform next (e.g. in patient care: asking a question or carrying out a test). Each of these steps has a cost and a benefit associated with it. The costs reflect monetary and/or non-monetary costs like the time necessary for conducting the step or the risk it bears. The benefit of a step depends on its utility on establishing or excluding currently considered diagnoses. Therefore the choice of the appropriate sequence of actions is mandatory for efficient diagnosis.

The sequence of actions to take can usually not be collected in the same formalism as the declarative knowledge, but is encoded at a lower level e.g. by numeric priorities reflecting the cost/benefit of a certain step. This breach in formalisms may result in a poor understandability and maintainability of the resulting knowledge base. Furthermore, simple linear functions may not be sufficient, if the right sequence of actions heavily depends on the outcome of previous tests.

To overcome these deficiencies, this paper proposes the formalization of diagnostic knowledge using the flowchart-based language *DiaFlux*. The resulting flows are an effective and intuitive knowledge representation to model diagnostic problem-solving. By combining well-known elements from flowcharts (i.e. nodes and transitions) with the tasks that are carried out during diagnostic problem-solving, it allows for a uniform and intuitive acquisition of declarative and procedural knowledge. As usually defined in flowcharts, the nodes represent steps, that have to be executed, and the transitions connecting those nodes the order of the execution. The transitions between nodes can be labelled with a condition. DiaFlux is intended for modelling diagnostic problem-solving, therefore the nodes are actions like performing a test or evaluating a diagnosis. The conditions attached to transitions are, e.g., the value of an already answered question or the evaluation of a diagnosis. Procedural knowledge necessary for problem solving in a given domain, that is formalized into a flowchart, is intuitively understandable and more easily maintainable. Besides the already mentioned goals, the following ones were pursued during the design of DiaFlux (inspired by [8]):

1. *Modularity*: To alleviate the reuse of parts of formalized knowledge, DiaFlux supports composed nodes, i.e. nodes that represent flowcharts themselves. The modularization also helps to improve the scalability of defined models.
2. *Repetitive execution of subtasks*: Online monitoring applications involve the continuous control of sensory data to diagnose fault situations and initiate corrective action. For this, particular actions need to be performed in an iterative manner.
3. *Parallelism*: Subtasks with no fixed order and dependency can be allocated to additionally spawned threads of control, and thus allow for their parallel execution. Expressing parallelism is especially necessary for mixed-initiative diagnosis of dynamic systems, in which human and machine initiated examinations are carried out concurrently.
4. *Representation of time*: DiaFlux is especially aimed at modelling knowledge for diagnosing dynamic processes. An integral part of the language is therefore a built-in representation of time and temporal reasoning. However, we omit a further discussion of this issue in the following of the paper.
5. *Testability*: The evaluation of a knowledge base (as it is for every software component) is an essential step prior to being put into productive use. We provide basic functionality for testing and analysing diagnostic flows.

The next section describes DiaFlux in general and shows how the above mentioned goals are fulfilled by it.

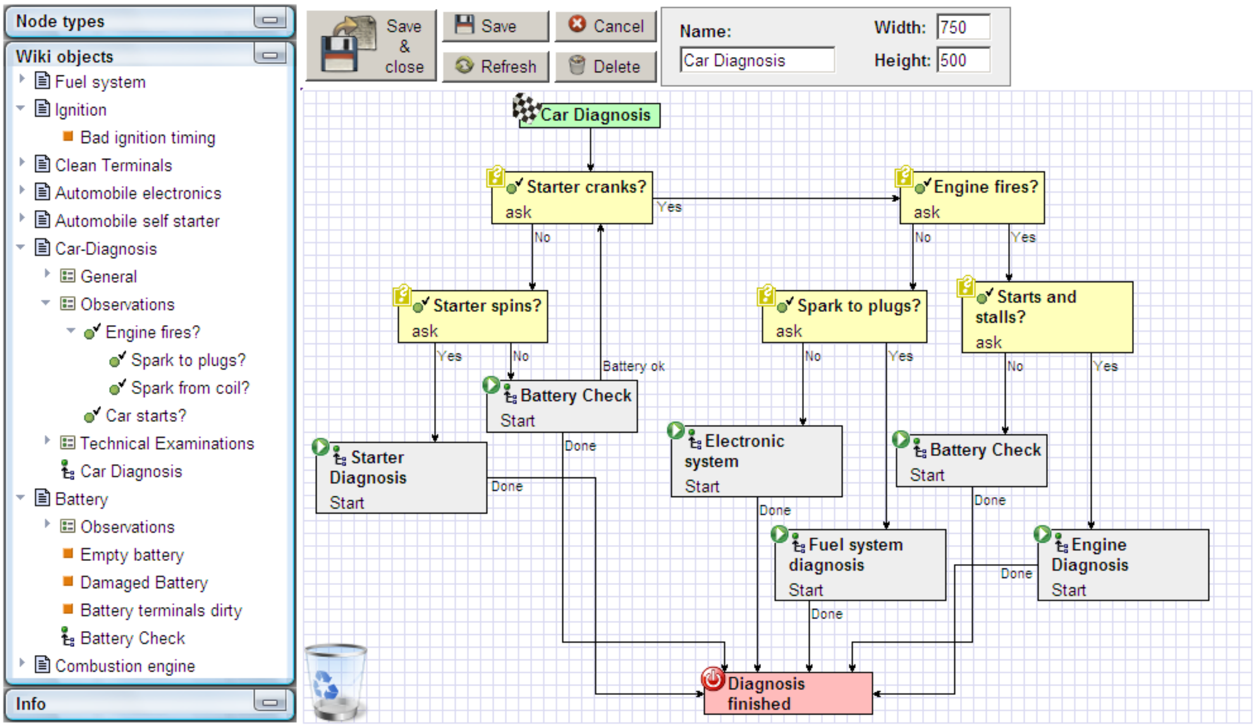


Figure 1: The diagnostic flow for a non-starting car, opened in the web-based editor of KnowWE

2 Modelling Diagnostic Flows with DiaFlux

The underlying formalism of DiaFlux being flowcharts, nodes and transitions possess the usual meaning. *Nodes* represent actions that have to be performed. *Transitions* control the possible sequences of actions, guided by the conditions that are associated with them. So, the procedural knowledge of “how” to conduct the overall task is mapped onto the transitions. Every node can have an arbitrary number of incoming and outgoing transitions.

Different node types are defined in DiaFlux, each representing a different type of action to take, when the node is activated during execution of the flowchart. Apart from different node types, there is only a single type of transition. However, various types of conditions can be modelled, depending on the type of node the transition starts at. As a diagnostic process is guided by the results of performed tests and the states of diagnoses, transitions are labelled with conditions that evaluate the status. The following node types are part of DiaFlux:

- *Start node:* A flowchart must have at least one starting node. More than one starting node per flowchart is supported for embedding different initial questionaires into a single diagnostic flow. As this construct might also be resembled by decomposing the flowchart and calling the resulting flowcharts in an appropriate way (as described in subsection 2.1), it does not enrich the expressivity of the language, but rather helps to avoid scattering knowledge belonging to a single scenario over several flowcharts and thus enhances the readability.
- *End node:* Each possible passage through the flowchart has to be terminatd by an *end node*. Different end nodes can be used to represent different outcomes of the process.
- *Test node:* Aimed at modelling diagnostic problem solving the main types of nodes in DiaFlux are directly derived from the kind of tasks, that have to be carried out during diagnostics, namely *performing tests* and *evaluating diagnoses*. A *test node* represents a single test, that is performed during runtime, when the according node is activated. This may trigger a questionnaire the user has to answer or information to be collected by sensors.
- *Diagnosis node:* The other main type is the *diagnosis node*. It allows altering the state of a diagnosis based on the observed symptoms.
- *Composed node:* A flowchart can reuse another flowchart by embedding it as a *composed node*. This results to a call to the embedded flowchart at runtime. Due to the possibility of multiple starting nodes, one starting node has to be explicitly selected for activation.
- *Fork node:* For expressing parallelism an explicit *fork node* allows spawning multiple concurrently running threads of control, called flows. These flows can later either be merged back at a *join node* or individually be terminated at an *end thread node*.
- *Join node:* For merging different flows back into a subsequent action, a join-node synchronizes the incoming flows and activates the subsequent action after all previously forked flows have reached it.
- *End thread node:* After spawning multiple flows each one can individually be terminated, if there is no common subsequent action to take by these flows later on. Terminating a single flow continues the execution of all other concurrently running flows.
- *Comment node*: For documentation purposes *comment nodes* can be inserted at arbitrary positions in the flowchart. These are ignored during execution.

- *Waiting node*: Diagnosing dynamic processes may involve a lapse of time, e.g. waiting for effects of curative action. For this reason, explicit *waiting nodes* can be inserted into the flowchart, that delay the execution of the subsequent actions. Having entered such a node, the execution is resumed after the defined period of time.
- *Loop node:* Using a loop node a fixed number of repetitions of a sequence of actions can be expressed.

In the following, we describe how the goals enumerated in Section 1 are fulfilled:

2.1 Modularity

The possibility to create modules of reusable knowledge is a prerequisite for the development of knowledge bases, that offer good maintanability. In a flowchart this property can be achieved in a natural way by the use of composed nodes. A composed node is a flowchart itself, that is processed when the according node is activated during runtime. As every flowchart can have multiple starting nodes, the call must contain the desired starting node of the called flowchart. After reaching an end node the execution is pointed back to the calling flowchart, evaluating the outgoing transitions of the node that triggered the call to determine the following action.

2.2 Repetitive execution

Monitoring most usually involves the repeated conduction of tests to measure the change of the diagnosed system over time. Besides cycles with fixed numbers of repetitions expressed by a loop node, repetitive execution of parts of a flowchart is supported by arbitrary cycles between nodes within a flowchart. By arbitrary cycles even unstructured loops with more than one entry and exit point can be modelled. For formulating loop conditions the full expressiveness of the underlying knowledge representation can be used.

2.3 Parallelism

For mixed-initiative monitoring and diagnosis in semi-closed loop systems parallel threads of execution are necessary. So, automatically conducted tests, as collecting sensory data, and user initiated ones can be carried out in parallel. For example, the sensory input can indicate a questionnaire that has to be answered by the user. Nevertheless, the diagnostic system has to be able to continue measuring sensory data and to trigger further actions concurrently (e.g. indicating another questionnaire or initiating corrective actions).

2.4 Representation of time

When diagnosing dynamic processes the underlying symptoms may change over time and hence also during the diagnosis itself. Tracking the trajectory of symptoms over time necessitates an explicit representation of time. This enables reasoning not only about the current values of symptoms but also about their progression. Having data temporally indexed allows furthermore for temporal abstractions for reasoning on a higher level.

2.5 Testability

Supporting the user during knowledge acquisition is an essential task, as the formalization of knowledge is an complex and error-prone task. DiaFlux, therefore, can ensure that certain properties hold for flowcharts, e.g. check for the disjointness and completeness of the conditions on outgoing transitions of every node. A more detailed discussion of this task is omitted for the rest of the paper.

2.6 Related Work

In [4] the use of production rules and event-graphs are proposed to represent declarative and procedural knowledge, respectively. Event-graphs, being bipartite graphs consisting of places and events together with a marking concept, strongly resemble Petri-Nets. When an event is activated it triggers the action that is associated with it. The knowledge base consists of two distinct databases that keep the production rules and the actions of the event-graphs separate. Thus, knowledge has to be duplicated if it is to be contained in both databases, reducing the overall maintainability. Production rules and event-graphs can interact both ways. The actions can modify the content of the rules' database and production rules may alter the markings of event-graphs.

Asbru [5] is a machine-readable language to represent time-oriented skeletal plans for reuse of procedural knowledge. They capture the essence of a procedure and are then, when applied, instantiated with distinct parameters and dynamically refined over time. The skeletal plans are to be executed by a human agent other than the original plan designer. For proper selection of plans, every plan specifies an intention it seeks to fulfill. Asbru is mainly used in the medical domain to implement clinical protocols, a more detailed version of clinical guidelines. Its focus lies more in the selection of therapy plans by their intentions, but less in the diagnostics of the underlying problem.

An approach for the collaborative development of clinical protocols in wikis can be found in [3]. Using predefined templates for the most frequent building blocks of Asbru, medical guidelines can easily be entered and exported.

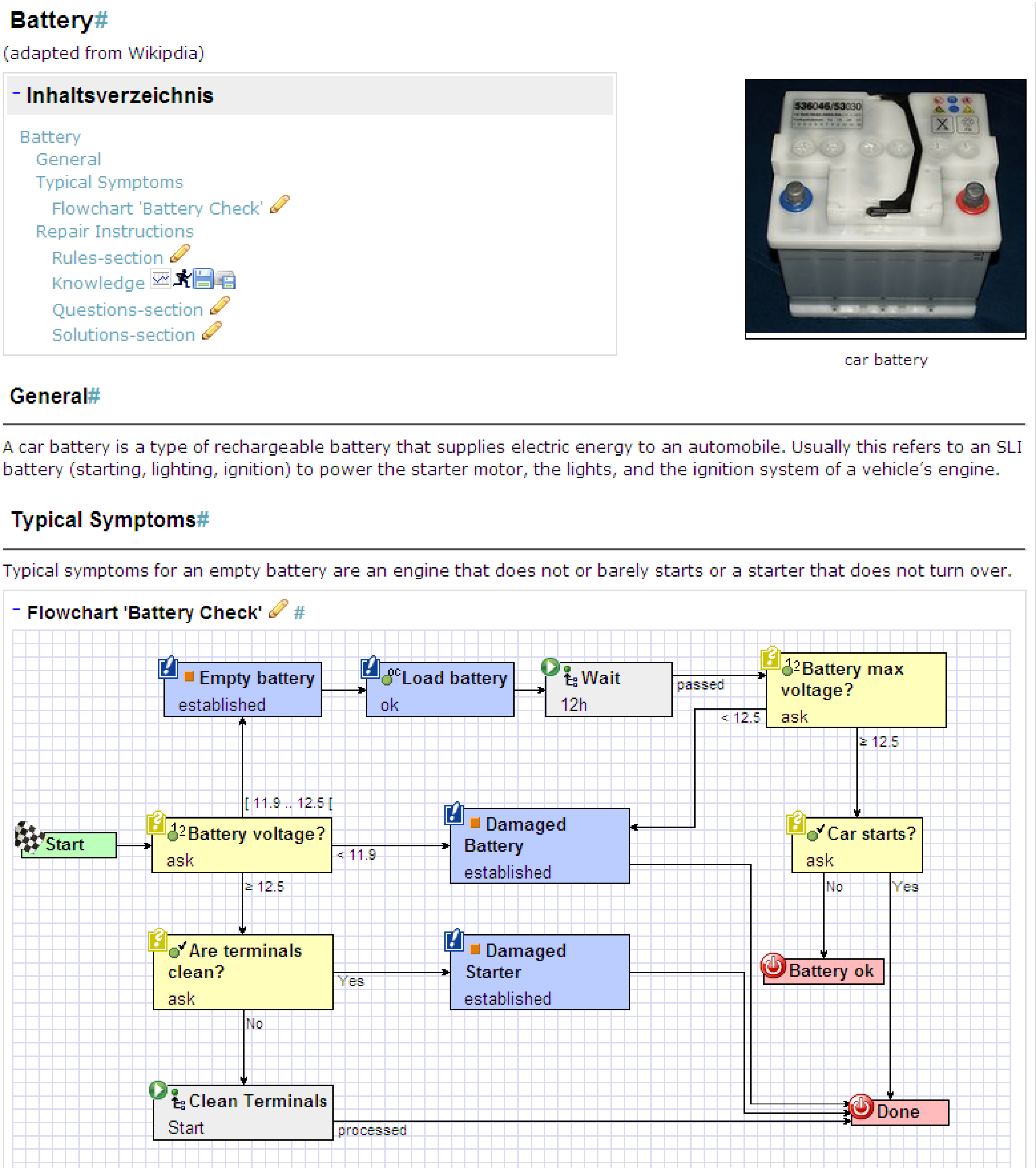
3 Implementation

The language DiaFlux introduced in the previous section has prototypically been implemented into the semantic wiki KnowWE [1]. By its flexible plugin mechanism KnowWE [7] can be extended to allow the creation of knowledge bases within a wiki using the knowledge representation of the knowledge-based system d3web. This approach for the collaborative development of knowlegde bases has shown its feasibility in several projects [6]. We intend to enable the collaborative development of flowcharts using DiaFlux. Hence we created a web-based flowchart editor, that is integrated into KnowWE via the beforementioned plugin mechanism.

Using the d3web-plugin for KnowWE every wiki article can contain a knowledge base declaring questions, diagnoses and rules. The DiaFlux-editor relies on knowledge elements that are readily available inside the wiki. For their simple reuse in a flowchart, the wiki is scanned for appropriate elements. These are then made available in a side-bar and can be dragged directly into the flowchart. Depending on the type of the node, different conditions can be formulated on the outgoing transitions.

At the moment the following node types for knowledge elements are supported:

- *Test*: When a test node is activated, the corresponding question is presented to the user. During execution, the process is not progressing until an answer for this question has been entered. In the conditions on outgo-



Battery#

(adapted from Wikipdia)

- Inhaltsverzeichnis

car battery

General#

A car battery is a type of rechargeable battery that supplies electric energy to an automobile. Usually this refers to an SLI battery (starting, lighting, ignition) to power the starter motor, the lights, and the ignition system of a vehicle's engine.

Typical Symptoms#

Typical symptoms for an empty battery are an engine that does not or barely starts or a starter that does not turn over.

- Flowchart 'Battery Check' #

Figure 2: Subtask for diagnosing an empty battery, embedded into a wiki article with additional information

ing nodes the possible answers to that question can be evaluated.

- *Diagnosis*: Inside a diagnosis node the status of a particular diagnosis can be modified. Processing a diagnosis node is instantly finished since no user interaction is necessary. As conditions the possible evaluations of a diagnosis by the user (solved, not solved) can be set. Depending on the user's agreement upon the diagnosis of the system, the process may end or continue.
- *Flowchart*: As mentioned earlier, already defined flowcharts can be reused as composed nodes within another flowchart. Since flowcharts can have more than one starting node, the call must contain the name of the desired starting node. The end node through which the called flowchart has been left can be considered as a return value and be used in the conditions of the transitions.

Figure 1 shows a flowchart in the web-based editor. It is implemented using Ajax and can therefore be used with every modern web-browser. The resulting flowchart is XML-encoded and embedded within a wiki article, enabling collaborative development.

4 Example: Car Diagnosis with DiaFlux

This section shows a small toy example for the diagnosis of a car, that does not properly start. It was modelled in DiaFlux. Though it does not exemplify all features of DiaFlux (e.g. no parallelism) it is fairly sufficient to present its main ideas.

First, the according declarative knowledge (tests and diagnoses, supplemented by additional informal information) has been entered into various knowledge bases inside the wiki articles using textual markups to define the basic terminology for car diagnosis [2]. The procedural diagnostic knowledge was then entered using the web-based editor of DiaFlux. Figure 1 shows the resulting diagnostic flow that mainly resembles a decision tree. It has a single starting node leading to the initial test node "Starter cranks?". At activation, the user is questioned if the car's starter is cranking. Since this is a yes/no-question, there are two outgoing transitions on this question node, each one labelled with one possible answer. If it is cranking, the starter is most likely working and the next question would be "Engine fires?", for which the user has to conduct further tests.

Assuming the starter is neither cranking nor spinning the composed node "Battery check" is reached, activating the starting node "Start" of the underlying diagnostic flow (shown in Figure 2). This subtask is a module for a battery testing procedure that is reused twice in the diagnostic flow. Its initial question "Battery voltage?" is a numerical question. Hence the outgoing transitions test for different possible voltage ranges. In this example the ranges are disjoint and complete, otherwise a warning would have been generated to inform the modeller. The first case tests if the voltage is below 11.9V. If it is, the battery is exhaustively discharged and considered broken. Therefore the solution "Battery damaged" is established, terminating the diagnostic process. If the battery's voltage lies in the range between [11.9V, 12.5V[it is probably too low for activating the starter, but can be recharged. The user is instructed to do so and wait for 12 hours. After this period has passed, the execution of the diagnostic flow is resumed, asking for the actual battery voltage. If the battery's maximum voltage is below 12.5V (considered as the minimum voltage for successfully starting a car), again the solution "Battery damaged" is established. For higher voltages, the user has to try starting the car. Depending on the outcome of this test, different end nodes are reached, closing the flow "Battery Check". After returning to the execution of the calling diagnostic flow, the conditions of the outgoing transitions of the composed node are checked. These test which of the exit nodes of the flow "Battery Check" has been taken. In case the car does not start, but its battery is working (end node "Battery ok"), a corresponding cyclic transition exists for returning to the inital test "Starter cranks?", restarting the diagnostic flow, which then will most likely lead to a solution other than "Empty Battery". In case the car starts (end node "Done"), the end node "Diagnosis finished" is activated, which terminates the entire diagnostic process. At higher battery voltages the reason for a non-starting car can be the battery terminals, that have to be tested next. If those are clean the solution "Damaged starter" is established, otherwise another module is called containing instructions of how to clean the terminals.

5 Conclusion

The development of (diagnostic) knowledge systems is a complex and time-consuming task. Usually, successful knowledge bases follow an optimized cost-benefit ratio, that take into account the costs of certain test steps in comparison to their benefit for the overall diagnostic process. *Diagnostic flows* is an intuitive language, that effectively combine considerations on the cost and the benefit of diagnostic processes. In this paper, we introduced the compact language DiaFlux, that can be used for the development of diagnostic flows. DiaFlux provides elementary node types for the modular, the parallel, and the time-oriented definition of diagnostic process knowledge. Besides a brief introduction into the language, we showed an implementation of DiaFlux within the semantic wiki KnowWE: Here, a plugin allows for the visual definition of flows in the wiki article. The use of a semantic wiki also opens the possibility to develop diagnostic flows in a collaborative manner. Also, the diagnostic process knowledge can be combined with tacit knowledge such as text and multimedia, that is also present in the wiki articles. The basic ideas of the language were demonstrated by a toy example taken from the car diagnosis domain. However, the language aims to be the basis for the collaborative development of a medical knowledge system in a running application project.

In the future, we consider the formulation of a framework of evaluation methods for the automated verification and validation of developed models. Further, it will become necessary to define appropriate refactoring methods, that support the evolution of the knowledge base. The impact of evaluation and evolution methods heavily depends on experiences, that we are going to make with real-world examples. Thus, we are currently planning to apply the introduced language to an extended protocol taken from the medical ICU domain.

References

[1] Joachim Baumeister and Frank Puppe. Web-based Knowledge Engineering using Knowledge Wikis. In *Proceedings of Symbiotic Relationships between Semantic Web and Knowledge Engineering (AAAI 2008 Spring Symposium)*, 2008.

[2] Joachim Baumeister, Jochen Reutelshoefer, and Frank Puppe. Markups for knowledge wikis. In *SAAKM'07: Proceedings of the Semantic Authoring, Annotation and Knowledge Markup Workshop*, pages 7–14, Whistler, Canada, 2007.

[3] Claudio Eccher, Marco Rospocher, Andreas Seyfang, Antonella Ferro, and Silvia Miksch. Modeling clinical protocols using Semantic MediaWiki: the case of the Oncocure project. In *ECAI 2008 Workshop on the Knowledge Management for Healthcare Processes (K4HelP)*, pages 20–24. University of Patras, 2008.

[4] Massimo Gallanti, Giovanni Guida, Luca Spampinato, and Alberto Stefanini. Representing procedural knowledge in expert systems: An application to process control. In *IJCAI*, pages 345–352, 1985.

[5] Silvia Miksch, Yuval Shahar, and Peter Johnson. Asbru: A task-specific, intention-based, and time-oriented language for representing skeletal plans. In *UK, Open University*, pages 9–1, 1997.

[6] Karin Nadrowski, Joachim Baumeister, and Volkmar Wolters. LaDy: Knowledge Wiki zur kollaborativen und wissensbasierten Entscheidungshilfe zu Umweltveränderung und Biodiversität. *Naturschutz und Biologische Vielfalt*, 60:171–176, 2008.

[7] Jochen Reutelshoefer, Florian Lemmerich, Fabian Haupt, and Joachim Baumeister. An extensible semantic wiki architecture. In *SemWiki'09: Fourth Workshop on Semantic Wikis – The Semantic Wiki Web*, 2009.

[8] B. Kiepuszewski W.M.P. van der Aalst, A.H.M. ter Hofstede and A.P. Barros. Workflow patterns. Technical report FIT-TR-2002-02, Queensland University of Technology, Brisbane, 2002.

Extraction of Adaptation Knowledge from Internet Communities*

Norman Ihle, Alexandre Hanft and Klaus-Dieter Althoff

University of Hildesheim
Institute of Computer Science
Intelligent Information Systems Lab
D-31141, Hildesheim, Germany
{ihle|hanft|althoff}@iis.uni-hildesheim.de

Abstract

[1] Acquiring knowledge for adaptation in CBR is an demanding task. This paper describes an approach to make user experiences from an Internet community available for the adaptation. We worked in the cooking domain, where a huge number of Internet users share recipes, opinions on them and experiences with them. Because this is often expressed in informal language, in our approach we did not semantically analyze those posts, but used our already existing knowledge model to find relevant information. We classified the comments to make the extracted and classified items usable as adaptation knowledge. The first results seem promising.

1 Introduction

Adaptation is a central part of the case-based reasoning process model [Kolodner, 1993]. A good adaptation is of very high importance if the case base is restricted to a small number of cases or if the variation of the problems to be solved is very high. Adaptation has not been the most current topic in recent years of CBR research [Greene *et al.*, 2008], but in the last year the research effort increased again [Cojan and Lieber, 2008; Cordier *et al.*, 2008; Leake and Dial, 2008]. Often adaptation means justifying values in a bounded range [Cojan and Lieber, 2008] and is done via rules created and maintained by a domain knowledge or system developer [Leake *et al.*, 1995].

Knowledge acquisition for adaptation (Adaptation Knowledge acquisition: AKA) is a cost intensive task since it is highly domain dependent and the hard-to-get experts are needed for acquiring and maintaining the necessary knowledge. To solve this problem, research on automated adaptation knowledge acquisition has been done, but mainly focused on automated AKA from cases in the case base [Wilke *et al.*, 1996; Hanney and Keane, 1997; d'Aquin *et al.*, 2007].

Besides the case base, the Internet and the Web 2.0 with its user-generated content are a large source of any kind of knowledge and experience. Following the Web 2.0 paradigm of user-interaction people provide their experience, opinions and advice on any kind of topic. Although the people are not necessarily experts in the domain, the hope is that the mass of users will correct mistakes as practiced for example in the Wikipedia project.

In this paper we present an approach to make knowledge from Internet communities accessible as adaptation knowledge using the domain model that usually exists in structured CBR applications [Bergmann, 2002]. In the first part of the paper the adaptation background inside CBR is introduced before we describe the domain and the existing application we worked with in our approach. After a short introduction of the tool we used, we will explain the approach in detail before we close with results of the evaluation, related work and an outlook.

2 The Cooking Domain

For most people cooking is ”everyday” knowledge. Almost everybody has encountered the problem of preparing a meal with a restricted amount of ingredients. This explains why a huge number of people are willing to share their recipes as well as their experience with the preparation of these recipes. Cooking communities on the Internet offer a platform for this. They provide the possibility to share recipes and also the chance to review and comment them. Usually they also offer additional information like cooking hints, background information on groceries or preparation methods. Besides the fact of the existence of active cooking communities, we chose the cooking domain for the investigation on adaptation knowledge because it has some advantages compared to other areas of interest discussed in communities like computer problems for example. First of all, it is relatively easy to describe the (near) complete context of preparing a meal. Hence, it is possible to reconstruct the experience of others by preparing the meal and trying the adaptation suggestions oneself. The context can be described according to the following characteristics:

1. all ingredients can be listed with exact amount and quality
2. ingredients can be obtained in standardized quantities and in comparable quality
3. kitchen machines and tools are available in a standardized manner
4. (in case of a failure) the preparation of a meal can start all over again every time from the same initial situation (except that we have more experience in cooking after each failure).

The latter one is not given in many other domains, for example setting up a computer costs much time and a certain installation situation cannot always be restored. Additionally, cooking and the adaptation of recipes is (some basic understanding presumed) relatively uncritical. In contrast to medical applications it does not endanger human health, except for some rare meals like fugu (pufferfish).

[1] This is an extended version of a paper which was accepted at the Workshop ”WebCBR: Reasoning from Experiences on the Web” at ICCBR'09

The costs of a failure are low. It is mostly covered by the price of the ingredients plus the preparation time. Cooking is also an appropriate application domain for adaptation, because cooking mastery depends on the variation and creativity, not only on following strictly preparation advices for a meal [This-Benckhard, 2001].

3 CookIIS and Adaptation in CookIIS

CookIIS [Hanft *et al.*, 2008] is a CBR-based recipe search engine that competes in the Computer Cooking Contest (CCC). When the user provides possible ingredients, it searches for suitable recipes in a case base. Doing that it considers ingredients the user does not want or cannot use because of a certain diet. If recipes with unwanted ingredients are retrieved, CookIIS offers adaptation suggestions to replace these ingredients. According to the CCC the set of recipes is restricted. Besides the retrieval and adaptation of recipes CookIIS also offers recipes for a complete three course menu from the given ingredients (and maybe some additional ones).

CookIIS is using a very detailed domain model which is described in [Hanft *et al.*, 2008]. It was created using the empolis:Information Access Suite (e:IAS) [empolis GmbH, 2008], which offers a knowledge modeling tool called Creator and with the Knowledge Server a component to build client-server based applications. It also provides a rule engine for the completion of cases and queries and for the adaptation of cases after the retrieval. Some more technical details are described in [Hanft *et al.*, 2009a].

3.1 Adaptation with the empolis:Information Access Suite

As stated above, the e:IAS offers the possibility to use completion rules which are executed before building the case index or before the retrieval to extend cases or queries with meta-information and adaptation rules, which are executed after the retrieval, to modify retrieved cases. The Creator offers a rule editor to model completion and adaptation rules with an own syntax. The rules follow the classic IF ... THEN ... schema. They have read and write access to all modeled objects and their values, but only adaptation rules have access to the retrieved cases since they are executed after the retrieval. A large amount of predefined functions help to manipulate single values. Both rule types use the same e:IAS specific syntax, which after compilation is stored in the format of the Orenge Rule Markup Language (ORML), an XML-language.

3.2 Case Representation and Similarity for Adaptation in CookIIS

The case representation is based on structured CBR. 11 classes of ingredients (e.g. Vegetables, Fruit, etc.) plus some classes for additional information (e.g. Type of Meal, Tools, etc.) are modeled, which represent about 2000 concepts of the cooking domain. As an example both concepts carrot and cucumber belong to the class Vegetable. The aim of differentiating ingredient classes is that we want to restrict the replacement of ingredients during adaptation to the same class. A case consists of 11 attributes, one for each possible ingredient class. Each attribute of ingredients can have multiple values per recipe (sets). Most concepts of the different classes are ordered in specific taxonomies. These and some custom similarity measures are used to compute the similarity between the query and the cases. Thereby the different attributes have different weights corresponding to their importance for a meal. Additional meta-information like the type of cuisine of a recipe is established during the indexing process of the recipes and also stored in the case.

The approach for adaptation that was first realized in CookIIS is to replace forbidden ingredients (according to a certain diet oder explicitly unwanted) with some similar ingredients of the same class. While executing a query unwanted (forbidden) ingredients are collected in extra attributes. Besides the explicit exclusion, four different methods can be distinguished to handle dietary practices, where more conditions have to be considered [Hanft *et al.*, 2009a]. One of those methods is the same approach as above: ingredients that have to be avoided due to a diet are replaced by similar ones.

Adaptation rules take these forbidden ingredients and check if at least one of them is an ingredient used in the retrieved case (recipe). Then, making use of the taxonomies and a similarity-aware set-function offered by the rule engine, the most similar ingredients to the unwanted one are retrieved and offered as replacement. The functions of the rules are described in detail in [Hanft *et al.*, 2009a]. If no similar ingredient is found that can be used for following the diet, then the suggestion is to just omit that ingredient.

Shortcomings of the Existing Adaptation Approach

Since the used adaptation approach makes use of the modeled taxonomies the results are often inappropriate. The method returns sibling concepts to the unwanted one as well as parent and child concepts. Only the siblings are the ones who are interesting for adaptation, but the others cannot be avoided with the provided rule functions. Also the number of siblings is often too high. For one unwanted ingredient one or two ingredients as an adaptation suggestion would be preferable. A detailed analysis of the problems with the adaptation and the ways to handle it with the e:IAS Rule mechanism is described in [Hanft *et al.*, 2009b].

4 CommunityCook: A System to Extract Adaptation Knowledge from Cooking Communities

In this chapter we will present our approach to extracting adaptation knowledge from a German cooking community. For this purpose we use our existing knowledge model from the CookIIS application and the TextMiner provided by e:IAS to extract ingredients from recipes and comments on those recipes and classify them. One of the classes can then be used as adaptation knowledge.

4.1 Idea behind the Approach

Our idea is to make knowledge from a cooking community accessible for our CookIIS application to have better adaptation suggestions in case a recipe contains an unwanted or forbidden ingredient. We were especially interested in comments that people posted in reply to provided recipes. In these comments users express their opinion on the recipe, before as well as after cooking it. They write about their experience with the preparation process and also tell what they changed while preparing the recipe. Thereby they express their personal adaptation of the recipe and frequently give reasons for this. Since this is written down in natural language text, often using informal language, we had the idea not to semantically analyze what people said, but to just find the occurrences of ingredients in the comment texts and then compare them to the ingredients mentioned in the actual recipe. We propose to classify them into three classes, depending on whether the ingredients mentioned in a comment appear in the recipe or not. The classification idea is described in the following sections.

4.2 Analysis of Example Cooking Communities

In Germany, *chefkoch.de*[2] is a well known cooking community with a large number of active users. So far, over 131'000 recipes have been provided by the users with an even larger amount of comments on them. The users also have the possibility to vote on the recipes, send them to a friend per email or even add pictures of their preparation. Besides the recipes, chefkoch.de features an open discussion board for all kinds of topics on cooking with more than 7.8 million contributions. Their English partner site *cooksunited.co.uk*[3] is unfortunately much smaller with only about 2200 recipes and 3500 posts.

But with *allrecipes.com*[4] a big platform with a huge amount of recipes and over 2.4 millions reviews is available in English. It has representable big localizations for the United States, Canada, the United Kingdom, Germany, France and others. Allrecipes.com explicitly provides variants of an existing recipe. Hence it also seems to be also a good source candidate. Another large cooking German community is *kochbar.de*[5] with over 160'000 recipes. Besides these large communities a number of smaller communities exist in the Web with more or less similar content. For our approach we decided to use a large German community since the recipes and the corresponding comments are presented on one page with a standardized HTML-code template, which makes it easier to crawl the site and extract relevant information items.

4.3 Extraction of Information Items from a Cooking Community

From a large German community we collected about 70'000 recipes with more than 280'000 comments by crawling the site. This way we got one HTML source-code page for each recipe with the corresponding comments. From this source code we extracted the relevant information entities using customized HTML-filters which we built using the HTML Parser tool[6]. For the recipes these entities were primarily the recipe title, needed ingredients and the preparation instructions, but also some additional information on the preparation of the recipe (e.g. estimated time for the preparation, difficulty of the preparation, etc.) and some usage statistics (e.g. a user rating, number of times the recipe has been viewed, stored or printed, etc.). If users commented on the recipe, we extracted the text of the comment, checked if the comment was an answer to another comment and if the comment has been marked as helpful or not. We also remembered the recipe ID of the related recipe. All this information we stored in a database to have an efficient access to it.

In the next step we used the e:IAS and indexed all recipes and all comments into two different case bases using a slightly extended CookIIS knowledge model. One case base consists of the recipes and one of the comments. For each recipe and each comment we extracted the mentioned ingredients and stored them in the case using our knowledge model and the e:IAS TextMiner during the indexing process. Since our knowledge model is bilingual (English and German) we were also able to translate the originally German ingredient names from the comment text into English terms during this process and this way had the same terms in the case bases that we use in our CookIIS application.

4.4 Classification of Ingredients

Having built up the two case bases we first retrieved a recipe and then all of the comments belonging to the recipe and compared the ingredients of the recipe with the ingredients mentioned in the comments. We then classified the ingredients mentioned in the comments into the following three categories:

- *New*: ingredients that are mentioned in the comment, but not mentioned in the recipe
- *Old*: ingredients that are mentioned in the comment as well as in the recipe
- *OldAndNew*: two or more ingredients of one class of our knowledge model, of which at least one was mentioned in the recipe and in the comment and at least one other one was only mentioned in the comment, but not in the recipe

We interpret the classification as follows:

- *New*: New ingredients are a variation of the recipe. A new ingredient (for example a spice or an herb) somehow changes the recipe in taste or is a tryout of something different or new.
- *Old*: If an ingredient of a recipe is mentioned in the comment it means that this ingredient is especially liked or disliked (for example the taste of it), that a bigger or smaller amount of this ingredient has been used (or even left out), or it is a question about this ingredient.
- *OldAndNew*: This is either an adaptation (e.g. instead of milk I took cream) or an explanation/specialization (e.g. Gouda is a semi-firm cheese).

For the adaptation the last class is the interesting one. For each ingredient classified as *OldAndNew* we also stored whether it is the new or the old one. We tried to distinguish between adaptation and specialization by looking for hints in the original comment text and by using the taxonomies of our knowledge model. Therefore we tried to find terms in the comment during the text-mining process that confirm if it is an adaptation (e.g. terms like: instead of, alternative, replaced with, ...) and stored those terms in the corresponding case. Additionally we looked in the taxonomy of the ingredient class whether the one ingredient is a child of the other (or the other way around). If an ingredient is a child of the other we interpreted this as specialization or explanation, because one ingredient is a more general concept than the other. This way we could avoid adaptations like: "instead of semi-firm cheese take Gouda".

For the classes *Old* and *New*, which we consider as variations of the recipe, we also tried to find terms in the comment that closer describe the function of the mentioned ingredient. For example, if an ingredient was classified as *Old*, we looked for terms like 'more', 'less' or 'left out'. If the ingredient of the comment is of the supplement class of our CookIIS knowledge model, and the recipe did not

[2] http://www.chefkoch.de, last visited 2009-04-22

[3] http://www.cooksunited.co.uk, last visited 2009-04-23

[4] http://allrecipes.com, last visited 2009-04-23

[5] http://www.kochbar.de, last visited 2009-05-22

[6] http://htmlparser.sourceforge.net, last visited 2009-04-18

contain any supplement, then we took this as a suggestion for a supplement (e.g. bread for a soup recipe).

For each classified ingredient we assigned a specific score, which depends on the following factors:

- the number of ingredients found in the comment text
- whether the comment was marked as helpful or not
- whether a term was found that indicates the classification assigned or not
- whether a term was found that indicates a different classification or not

After assigning the score we aggregated our classification results. We did this in two steps: First we aggregated all classified ingredients of all comments belonging to one recipe. Thereby we counted the number of the same classifications in different comments and added up the score of the same classifications. For instance a specific recipe has 12 comments in which 3 of them mention milk and cream.

Then we aggregated all classifications without regarding the recipe they belong to. In our dataset we found comments with milk and cream belonging to 128 different recipes. This way we could select the most common classifications out of all classifications. Since we are using a CBR tool and have cases, we also checked if similar recipes have the same ingredients with the same classification mentioned in the comments. We did this for each recipe first with a similarity of at least 0.9, then with a similarity of 0.8. If many of the same classified ingredients exist in similar recipes, this supports our results.

4.5 Usage as Adaptation Knowledge

OldAndNew-classified ingredients can be used to generate adaptation suggestions. This can be done in two different ways: independent from the recipe or with regard to the recipe. Considering the fist way, we look in the database table for the ingredient to adapt and use the result where the ingredient that needs to be adapted is categorized as old and appears in the most recipes or has the highest score. It is possible to retrieve two or more adaptation suggestions to be more manifold. Using this approach we got more than 6200 different adaptation suggestions of which we only used the most common (regarding the number of appearances in the comments and the score) per ingredient. Figure 1 shows some of these suggestions, e.g. in the first line a suggestion to replace cream with milk which appears in comments to 128 different recipes.

We integrated this approach into our CookIIS application: at first we look for two adaptation suggestions from CommunityCook. If no suggestions are provided, the set of more general adaptation rules (see section 3.2) determine adaptation suggestions.

5 Evaluation of the Results

A first look at the results of the most common adaptation suggestions is promising. Only the ingredient class ”supplement” reveals problems which are due to the fact that too many different ingredients are integrated into this class. This can be changed by further improving the modeling.

The evaluation can be divided into two different parts. At first we checked if our classification and the interpretation correspond to the intentions written in the original comments. This was done manually by comparing the classification results and their interpretation to the original comments and match in most of over 400 tests the classification.

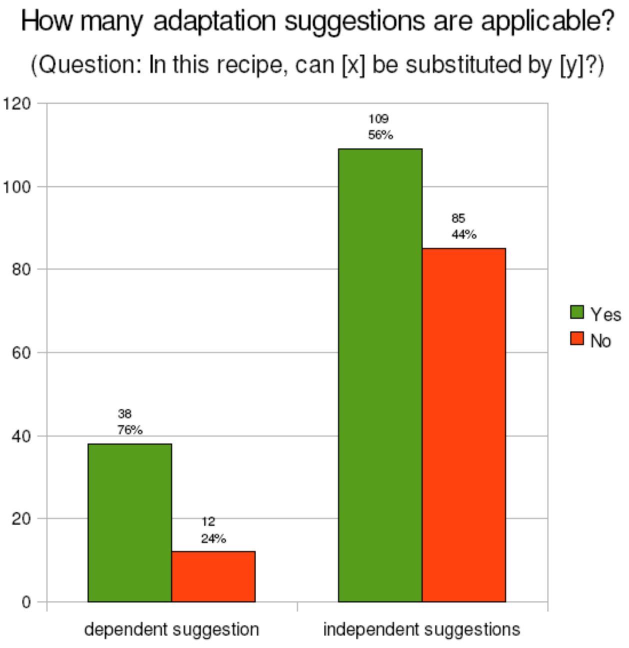


Figure 2: Applicability of dependent and overall independent suggestion

The second evaluation was done on the results of the overall aggregated adaptation suggestions. We examined whether the adaptation suggestions with a high score are good adaptation suggestions for any kind of recipe. Our idea is to take a representative number of recipes and present them with adaptation suggestions to real chefs. These chefs then rate the adaptation suggestions.

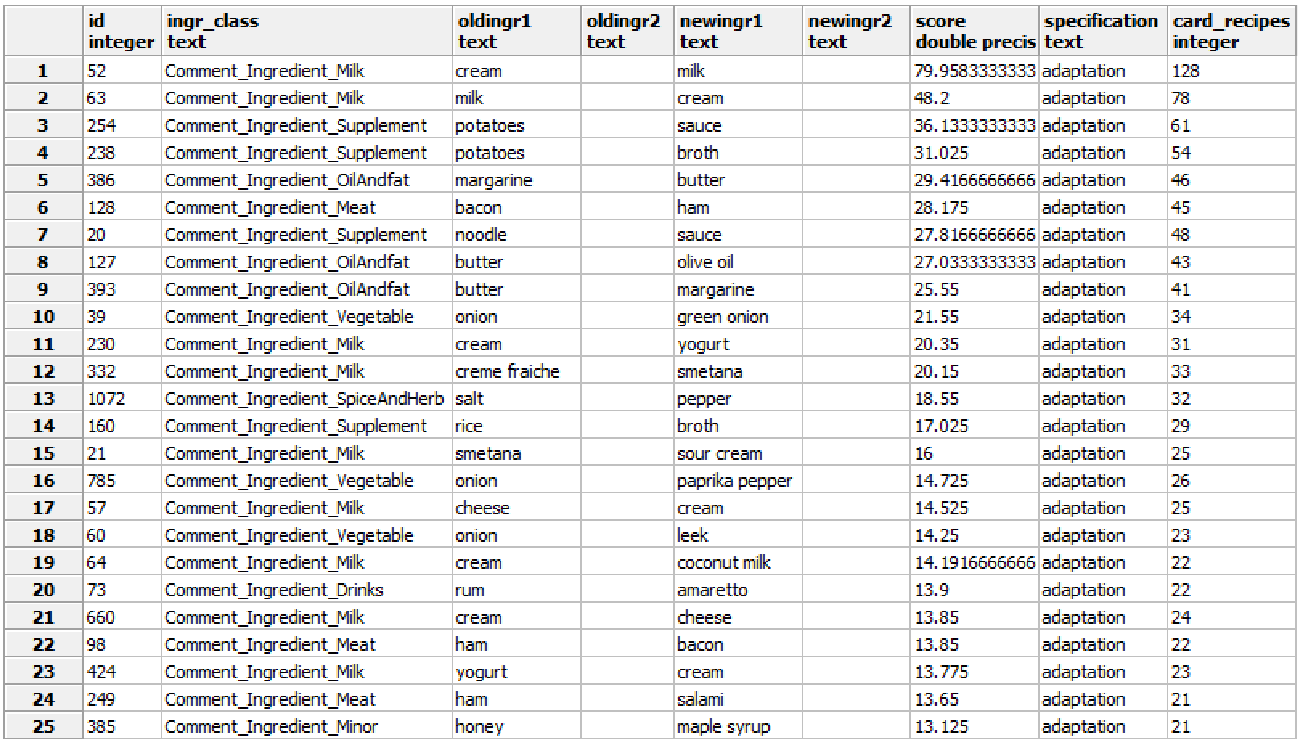
Therefore we designed a questionnaire by choosing randomly a recipe and add one adaptation suggestion extracted from comments belonging to that recipe (”dependent”) and secondly add two adaptation suggestions without regard of the recipe (”independent”) each with two ingredients as replacement suggestion. At the end we present the 50 questionnaires with 50 dependent and 100 pairs of independent ingredients to different chefs, because each chef may have a different opinion.

76% of the dependent and 58% of the independent adaptation suggestions were confirmed as applicable by the chefs (see figure 2). Differentiating the first and second independent suggestion it could be observed that the first one is noteworthy better (see figure 3). Summing up it follows that only by 11 of the 100 independent adaptation suggestions no ingredient can be used as substitution.

In case that the adaptation suggestion was applicable, the chefs should rate it as very good, good and practicable. Here again the dependent suggestions perform better, see figure 4.

6 Related Work

JULIA [Hinrichs, 1992] and CHEF [Hammond, 1986] are early CBR systems giving preparation advice for meals. CHEF is a planning application which builds new recipes in the domain of Szechwan cooking. To satisfy the goals of a request for a new recipe it anticipates and tries to avoid problems. Therefore it stores and retrieves occurred problems and ways of dealing with them. JULIA integrates CBR and constraints for menu design tasks. It uses a large taxonomy of concepts and problem decomposition with fixed decomposition plans. Unlike our approach their



	id integer	ingr_class text	oldingr1 text	oldingr2 text	newingr1 text	newingr2 text	score double precis	specification text	card_recipes integer
1	52	Comment_Ingredient_Milk	cream		milk		79.9583333333	adaptation	128
2	63	Comment_Ingredient_Milk	milk		cream		48.2	adaptation	78
3	254	Comment_Ingredient_Supplement	potatoes		sauce		36.1333333333	adaptation	61
4	238	Comment_Ingredient_Supplement	potatoes		broth		31.025	adaptation	54
5	386	Comment_Ingredient_OilAndfat	margarine		butter		29.4166666666	adaptation	46
6	128	Comment_Ingredient_Meat	bacon		ham		28.175	adaptation	45
7	20	Comment_Ingredient_Supplement	noodle		sauce		27.8166666666	adaptation	48
8	127	Comment_Ingredient_OilAndfat	butter		olive oil		27.0333333333	adaptation	43
9	393	Comment_Ingredient_OilAndfat	butter		margarine		25.55	adaptation	41
10	39	Comment_Ingredient_Vegetable	onion		green onion		21.55	adaptation	34
11	230	Comment_Ingredient_Milk	cream		yogurt		20.35	adaptation	31
12	332	Comment_Ingredient_Milk	creme fraiche		smetana		20.15	adaptation	33
13	1072	Comment_Ingredient_SpiceAndHerb	salt		pepper		18.55	adaptation	32
14	160	Comment_Ingredient_Supplement	rice		broth		17.025	adaptation	29
15	21	Comment_Ingredient_Milk	smetana		sour cream		16	adaptation	25
16	785	Comment_Ingredient_Vegetable	onion		paprika pepper		14.725	adaptation	26
17	57	Comment_Ingredient_Milk	cheese		cream		14.525	adaptation	25
18	60	Comment_Ingredient_Vegetable	onion		leek		14.25	adaptation	23
19	64	Comment_Ingredient_Milk	cream		coconut milk		14.1916666666	adaptation	22
20	73	Comment_Ingredient_Drinks	rum		amaretto		13.9	adaptation	22
21	660	Comment_Ingredient_Milk	cream		cheese		13.85	adaptation	24
22	98	Comment_Ingredient_Meat	ham		bacon		13.85	adaptation	22
23	424	Comment_Ingredient_Milk	yogurt		cream		13.775	adaptation	23
24	249	Comment_Ingredient_Meat	ham		salami		13.65	adaptation	21
25	385	Comment_Ingredient_Minor	honey		maple syrup		13.125	adaptation	21

Figure 1: Some suggestions for adaptation

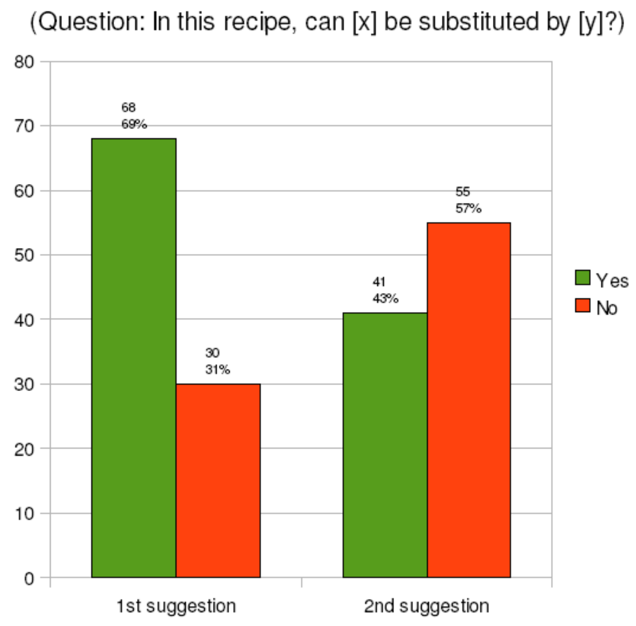


Figure 3: Applicability of the first and second independent suggestion

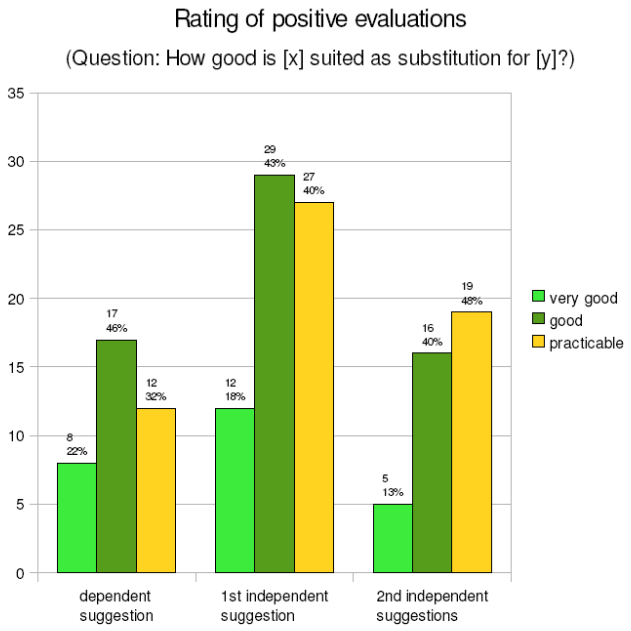


Figure 4: Ratings of applicable adaptation suggestions

knowledge was built by experts and was not captured from communities.

The idea presented here closely relates to the research of Plaza [Plaza, 2008], especially the EDIR cycle, however they concentrate more on gathering cases from web experience. In [Cordier *et al.*, 2008] they use the presented IakA approach for the acquisition of adaptation knowledge (and cases) by asking an oracle, which is described as an "ideal expert", but the presented prototype IakA-NF works (only) for numerical function domains. Furthermore Acquisition of Adaptation Knowledge from cases was done by by [Hanney and Keane, 1997] or with the CABAMAKA System by [d'Aquin *et al.*, 2007].

The procedure of looking at first for concrete adaptation suggestions and apply afterwards, if the first step yields no results, more general rules, was done also by [Leake *et al.*, 1995] with DIAL, which at first attempt to retrieve adaptation cases.

Our approach presented here goes with the vision of Collaborative Multi-Experts Systems (CoMES) [Althoff *et al.*, 2007] and is modelled following the SEASALT architecture [Bach *et al.*, 2007], an instance of CoMES. Mapping this to the CommunityCook System the collection of recipes and comments corresponds to the task of the *Collector Agent*. The further analysis and interpretation match to their role of a *Knowledge Engineer*.

7 Conclusion and Outlook

Adaptation knowledge acquisition is an demanding and expensive task since it needs experts. In this paper we presented an approach to use experience from Internet communities for adaptation knowledge. Our approach is based on the idea of comparing the ingredients mentioned in a recipe to the ones mentioned in the comments that relate to the recipe. From comments which contain ingredients also existing in the recipe and others which are not contained in the recipe the adaptation suggestions are created and aggregated over all comments to 6200 suggestions. The evaluation results are promising and show that adaptation suggestion extracted from the same recipe are more acceptable than the one which are independent and aggregated over all recipes.

The approach described here has a lot of advantages. For finding ingredients we can use our existing CookIIS knowledge model which has the benefit of taking care of synonyms, independence from slang and grammatically deficient language. By using a large number of recipes and comments we hope to balance out wrong classifications. We integrated the extracted adaptation suggestions in our CookIIS application.

In the future we want to be able to use the adaptation suggestions with regard to the recipe they belong to. Therefore we will find similar recipe out of our pool of 70'000 recipes to the one that has to be adapted and consider only comments of these recipes following the principle that similar recipes need similar adaptations.

Following the SEASALT architecture we also want to realize a multi-agent system that continuously monitors the community for new experiences with the recipes and adapts our adaptation knowledge if necessary.

References

[Althoff *et al.*, 2007] Klaus-Dieter Althoff, Kerstin Bach, Jan-Oliver Deutsch, Alexandre Hanft, Jens Mänz, Thomas Müller, Régis Newo, Meike Reichle, Martin Schaaf, and Karl-Heinz Weis. Collaborative multi-expert-systems - realizing knowledge-lines with case factories and distributed learning systems. In Joachim Baumeister and Dietmar Seipel, editors, *KESE*, volume 282 of *CEUR Workshop Proceedings*, 2007.

[Althoff *et al.*, 2008] Klaus-Dieter Althoff, Ralph Bergmann, Mirjam Minor, and Alexandre Hanft, editors. *Advances in Case-Based Reasoning, 9th European Conference, ECCBR 2008, Trier, Germany, September 1-4, 2008. Proceedings*, volume 5239 of *LNCS*, Heidelberg, 2008. Springer.

[Bach *et al.*, 2007] Kerstin Bach, Meike Reichle, and Klaus-Dieter Althoff. A domain independent system architecture for sharing experience. In Alexander Hinneburg, editor, *Proceedings of LWA 2007, Workshop Wissens- und Erfahrungsmanagement*, pages 296–303, September 2007.

[Bergmann, 2002] Ralph Bergmann. *Experience Management: Foundations, Development Methodology, and Internet-Based Applications*, volume 2432 of *LNAI*. Springer-Verlag, 2002.

[Cojan and Lieber, 2008] Julien Cojan and Jean Lieber. Conservative adaptation in metric spaces. In Althoff et al. [2008], pages 135–149.

[Cordier *et al.*, 2008] Amélie Cordier, Béatrice Fuchs, Léonardo Lana de Carvalho, Jean Lieber, and Alain Mille. Opportunistic acquisition of adaptation knowledge and cases - the iaka approach. In Althoff et al. [2008], pages 150–164.

[d'Aquin *et al.*, 2007] Mathieu d'Aquin, Fadi Badra, Sandrine Lafrogne, Jean Lieber, Amedeo Napoli, and Laszlo Szathmary. Case base mining for adaptation knowledge acquisition. In Manuela M. Veloso, editor, *IJCAI*, pages 750–755. Morgan Kaufmann, 2007.

[empolis GmbH, 2008] empolis GmbH. Technical white paper e:information access suite. Technical report, empolis GmbH, January 2008.

[Greene *et al.*, 2008] Derek Greene, Jill Freyne, Barry Smyth, and Pádraig Cunningham. An analysis of research themes in the cbr conference literature. In Althoff et al. [2008], pages 18–43.

[Hammond, 1986] Kristian J. Hammond. Chef: A model of case-based planning. In *American Association for Artificial Intelligence, AAAI-86, Philadelphia*, pages 267–271, 1986.

[Hanft *et al.*, 2008] Alexandre Hanft, Norman Ihle, Kerstin Bach, Régis Newo, and Jens Mänz. Realising a cbr-based approach for computer cooking contest with e:ias. In Martin Schaaf, editor, *ECCBR Workshops*, pages 249–258, Hildesheim, Berlin, 2008. Tharax Verlag.

[Hanft *et al.*, 2009a] Alexandre Hanft, Norman Ihle, Kerstin Bach, and Régis Newo. Cookiis – competing in the first computer cooking contest. *Künstliche Intelligenz*, 23(1):30–33, 2009.

[Hanft *et al.*, 2009b] Alexandre Hanft, Norman Ihle, and Régis Newo. Refinements for retrieval and adaptation of the cookiis application. In Knut Hinkelmann and Holger Wache, editors, *Wissensmanagement*, volume 145 of *LNI*, pages 139–148. GI, 2009.

[Hanney and Keane, 1997] Kathleen Hanney and Mark T. Keane. The adaptation knowledge bottleneck: How to ease it by learning from cases. In David B. Leake and Enric Plaza, editors, *ICCBR*, volume 1266 of *LNCS*, pages 359–370. Springer, 1997.

[Hinrichs, 1992] Thomas R. Hinrichs. *Problem solving in open worlds*. Lawrence Erlbaum, 1992.

[Kolodner, 1993] Janet L. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, San Mateo, 1993.

[Leake and Dial, 2008] David B. Leake and Scott A. Dial. Using case provenance to propagate feedback to cases and adaptations. In Althoff et al. [2008], pages 255–268.

[Leake *et al.*, 1995] David B. Leake, Andrew Kinley, and David C. Wilson. Learning to improve case adaptation by introspective reasoning and cbr. In Manuela M. Veloso and Agnar Aamodt, editors, *ICCBR*, volume 1010 of *LNCS*, pages 229–240. Springer, 1995.

[Plaza, 2008] Enric Plaza. Semantics and experience in the future web. In Althoff et al. [2008], pages 44–58. invited talk.

[This-Benckhard, 2001] Herve This-Benckhard. *Rätsel und Geheimnisse der Kochkunst*. Piper, 2001.

[Wilke *et al.*, 1996] Wolfgang Wilke, Ivo Vollrath, Klaus-Dieter Althoff, and Ralph Bergmann. A framework for learning adaptation knowledge based on knowledge light approaches. In *5th German Workshop on CBR*, pages 235–242, 1996.

What you *understand* is what you get: Assessment in Spreadsheets

Andrea Kohlhase and Michael Kohlhase
German Center for Artificial Intelligence (DFKI)
{Andrea,Michael}.Kohlhase@dfki.de

Abstract

Spreadsheets are heavily employed in administration, financial forecasting, education, and science because of their intuitive, flexible, and direct approach to computation. In previous work we have studied how an explicit representation of the background knowledge associated with the spreadsheet can be exploited to alleviate usability problems with spreadsheet-based applications. The SACHS system implements this approach to provide a semantic help system for DCS, an Excel-based financial controlling system.

In this paper, we evaluate the coverage of the SACHS system with a "Wizard of Oz" experiment and see that while SACHS fares much better than DCS alone, it systematically misses important classes of explanations. We provide a first approach for an "assessment module" in SACHS. It assists the user in judging the situation modeled by the data in the spreadsheets and possibly remedying shortcomings.

1 Introduction

Semantic technologies like the Semantic Web promise to add novel functionalities to existing information resources adding explicit representations of the underlying objects and their relations and exploiting them for computing new information. The main intended application of the Semantic Web is to combine information from various web resources by identifying concepts and individuals in them and reasoning about them with background ontologies that make statements about these.

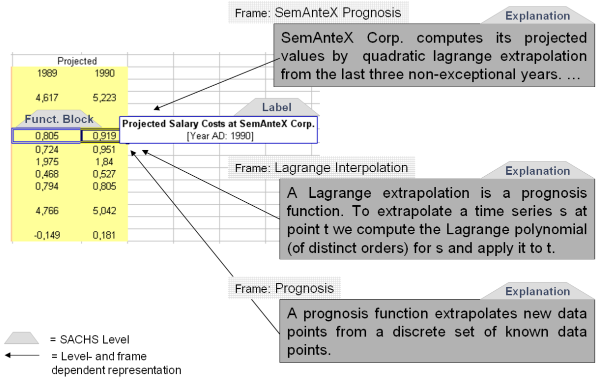
We follow a different, much less researched approach here. We call it **Semantic Illustration**: Instead of enhancing web resources into semi-formal ontologies[1] by annotating them with formal objects that allow reasoning as in the Semantic Web paradigm, the Semantic Illustration architecture *illustrates* a software artifact with a semi-formal ontology by complementing it with enough information to render new semantic services (compare to a somewhat analogous requirement phrased in [Tag09]).

Concretely, in the SACHS system [KK09a] we provide a semantic help system for "**DCS**", a financial controlling system based on Excel [Mic] in daily use at the German Center for Artificial Intelligence (DFKI). Here we illustrate a spreadsheet with a semi-formal ontology of the relevant background knowledge via an interpretation mapping. Then we use the formal parts of the ontology to control the aggregation of help texts (from the informal part of the ontology) about the objects in the spreadsheet. This enables in turn new semantic interaction services like "semantic navigation" or "framing" (see [KK09c]).

There are other instances of the Semantic Illustration paradigm. In the CPoint system (e.g. [Koh07]), the objects of a MS PowerPoint presentation are complemented with information about their semantic status, and this information is used for eLearning functionalities. In the FormalVi system [KLSS09], CAD/CAM developments are illustrated with formal specifications, so that safety properties of the developments can be verified and agile development of robot parts can be supported by tracing high-level design requirements and detecting construction errors early. Finally, semantic technologies like the "Social Semantic Desktop" (see e.g. [SGK$^+$06]) fit into the Semantic Illustration approach as well, since they complement software artifacts in the computer desktop (e-mails, contacts, etc.) with semantic information (usually by letting the user semantically classify and tag them) and use the semantic structure to enhance user interaction.

With the SACHS system in a usable state, we have started evaluating it with respect to user acceptance and coverage. To keep the paper self-contained we give a short overview of the SACHS system in the next section, followed by the coverage evaluation experiment in Section 3. This reveals that the DCS system only models the factual part of the situation it addresses, while important aspects for 'understanding the numbers' remain implicit — and as a consequence the SACHS system also fails to tackle them. For instance, users often ask questions like "*Is it good or bad if this cell has value 4711?*" and experienced controllers may tell users "*Cell D16 must always be higher than E5*". We consider this knowledge (which we call **assessment knowledge**) to be an essential part of the background knowledge to be modeled in the semantically enhanced spreadsheet systems, since a person can only profit from help if it is understood in (all) its consequences. In particular, the assessment knowledge must be part of the user assistance part (e.g. answering the first

[1] With this we mean ontologies with added documentation ontologies so that they can be read by non-experts or texts annotated with ontological annotations either by in-text markup or standoff-markup.

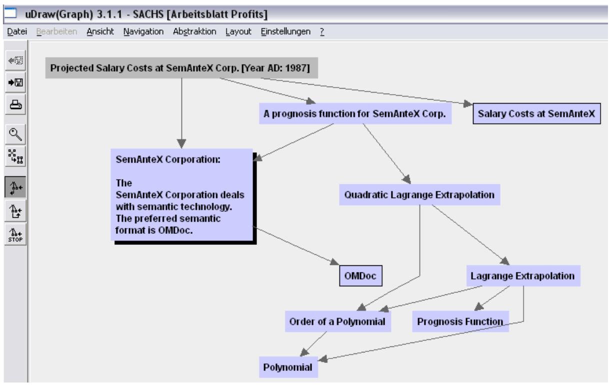


SemAnteX Corp. computes its projected values by quadratic lagrange extrapolation from the last three non-exceptional years. …

A Lagrange extrapolation is a prognosis function. To extrapolate a time series s at point t we compute the Lagrange polynomial (of distinct orders) for s and apply it to t.

A prognosis function extrapolates new data points from a discrete set of known data points.

Figure 2: Explanations within Distinct Frames



The
SemAnteX Corporation deals
with semantic technology.
The preferred semantic
format is OMDoc.

Figure 3: Dependency Graph with ’uses’-Edges

question) and can be used to issue warnings (e.g. if the controller’s invariant inherent in the second statement is violated).

We will present a preliminary approach for modelling the background knowledge involved in assessment in Section 4 and envision how this can be used in the SACHS system in Section 5. Section 6 concludes the paper and discusses future research directions.

2 SACHS (Semantic Annotation for a Controlling Help System)

For SACHS we took a foundational stance and analyzed spreadsheets as semantic documents, where the formula representation is the computational part of the semantic relations about how values were obtained. To compensate the diagnosed computational bias (published in [KK09a]) we augmented the two existing semantic layers of a spreadsheet — the surface structure and the formulae — by one that makes the intention of the spreadsheet author explicit.

The central concept we establish is that of a **functional block** in a spreadsheet, i.e., a rectangular region in the grid where the cells can be interpreted as input/output pairs of a *function* (the **intended function** of the functional block). For instance, the cell range [B17:F17] in Figure 1[2] (highlighted with the selection of [B17] by a borderline) is a functional block, since the cells represent profits as a function π of time; the pair $\langle 1984, 1.662\rangle$ of values of the cells [B4] and [B17] is one of the pairs of π.

The semantic help functionality of the SACHS system is based on an **interpretation** mapping, i.e., a meaning-giving function that maps functional blocks to concepts in a background ontology. For instance our functional block [B17:F17] is interpreted to be the function of “Actual Profits at SemAnteX Corp.” which we assume to be available in the semantic background.

In [KK09a] we have presented the SACHS information and system architecture, and have shown how the semantic background can be used to give semantic help to the user on several levels like labels, explanations (as showcased in Figure 2) and dependency graphs like the one for cell [G9] in Figure 3. This graph-based interface allows the user to navigate the structured background ontology by definitional structure of intended functions. In this case the graph also reveals that the spreadsheet concerns the profit statement of the business “SemAnteX Corp.”, which is not represented in the spreadsheet alone.

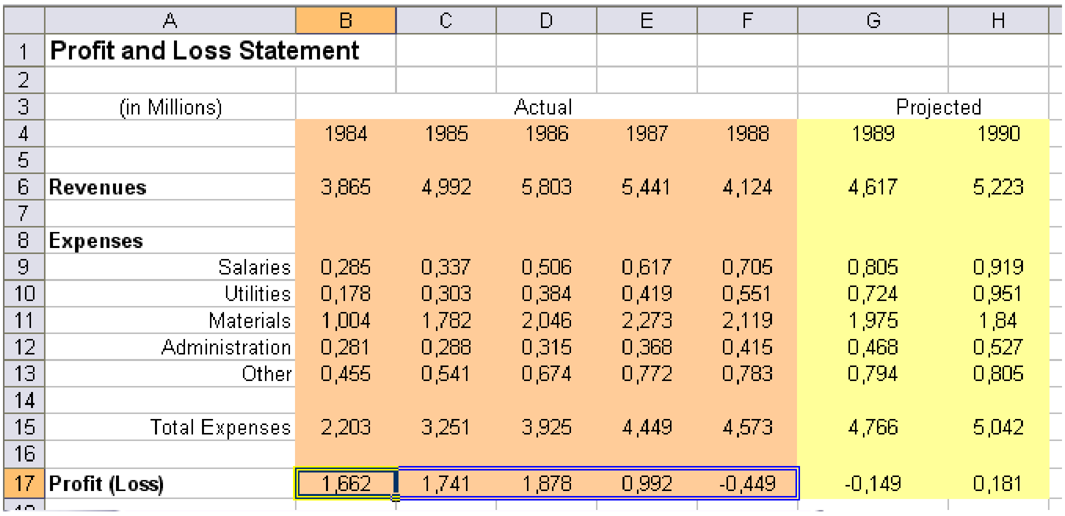
While the information about functional blocks and the meaning of their values (e.g. units), the provenance of data, and the meaning of formulae provided by the semantic background are important information, the development process made it painfully clear that the interpretation (hence the information provided by the SACHS system to the user) is strongly dependent on the author’s point of view — how she *frames* the data. We have developed a first **theory of framing** based on theory-graphs and theory morphisms in [KK09c], and have extended the interaction based on this. Among others, this enables the SACHS system to (i) tailor the help texts to the frame chosen by the user (and thus presumably to the task she pursues; see three distinct explanations in Figure 2), and (ii) to provide frame alternatives for exploring the space of possible **spreadsheet variants** e.g. for different prognosis scenarios.

3 Help Needed, but Where?

To develop the theory graph for the background knowledge of the DFKI Controlling system we organized interviews with a DFKI expert on the topic and recorded them as MP3 streams[3]. Even though these interviews were not originally intended as a “Wizard of Oz” experiment, in the following we will interpret them so. A **“Wizard of Oz” experiment** is a research experiment in which subjects interact with a computer system that subjects believe to be autonomous, but which is actually being operated or partially operated

[2] This spreadsheet is our running example, also taken up in Section 4.

[3] We recorded three interview sessions amounting to approximately 1.5 hrs concerning 39 distinct knowledge items and containing 110 explanations. Naturally, there were more informal question and answer sessions mostly by email or phone afterwards, but we cannot take these into account here unfortunately. In hindsight we realize that we should have annotated the interviews contained many “references by pointing”, which are lost in the recording. For instance, in the specific spreadsheet the legend for various cells are very specific like “linearised contract volume with pass-through” and “linearised contract volume without pass-through”. When talking about the cells both are abbreviated to “linearised contract volume” and which cell is really talked about is pointed at with fingers leaving the interest listener with wonder.



	A	B	C	D	E	F	G	H
1	**Profit and Loss Statement**							
2								
3	(in Millions)	Actual					Projected	
4		1984	1985	1986	1987	1988	1989	1990
5								
6	**Revenues**	3,865	4,992	5,803	5,441	4,124	4,617	5,223
7								
8	**Expenses**							
9	Salaries	0,285	0,337	0,506	0,617	0,705	0,805	0,919
10	Utilities	0,178	0,303	0,384	0,419	0,551	0,724	0,951
11	Materials	1,004	1,782	2,046	2,273	2,119	1,975	1,84
12	Administration	0,281	0,288	0,315	0,368	0,415	0,468	0,527
13	Other	0,455	0,541	0,674	0,772	0,783	0,794	0,805
14								
15	Total Expenses	2,203	3,251	3,925	4,449	4,573	4,766	5,042
16								
17	**Profit (Loss)**	1,662	1,741	1,878	0,992	-0,449	-0,149	0,181

Figure 1: A Simple (Extended) Spreadsheet after [Win06]

by an unseen human being (see [Wik09]). Here, the interviewee plays the part of an ideal SACHS system and gives help to the interviewer who plays the part of the user. This experiment gives us valuable insights about the *different qualities of knowledge* in a user assistance system, which the expert thought was necessary to understand the specific controlling system spreadsheet.

When studying the MP3 streams, we were surprised that in many cases a question of "*What is the meaning of ...*" was answered by the expert with up to six of the following **explanation types**, the occurrence rate of which relative to the number of knowledge items is listed in the brackets:

1. **Definition (Conceptual)** [71.8%]

   A *definition* of a knowledge item like a functional block is a thorough description of its meaning. For example the functional block "cover ratio per project in a research area" was defined as the percentage rate to which the necessary costs are covered by the funding source and own resources.

2. **Purpose (Conceptual)** [46.2%]

   The *purpose* of a knowledge item in a spreadsheet is defined by the spreadsheet author's intention, in particular, the purpose explains why the author put the information in. A principal investigator of a project or the respective department head e.g. needs to get the information about its cover ratio in order to know whether either more costs have to be produced to exploit the full funding money or more equity capital has to be acquired.

3. **Assessment of Purpose** [30.8%]

   Given a purpose of a knowledge item in a spreadsheet, its reader must also be able to reason about the purpose, i.e., the reader must be enabled to draw the intended conclusions/actions or to *assess the purpose*. For understanding whether the cover ratio is as it is because not enough costs have yet been produced, the real costs have to be compared with the necessary costs. If they are still lower, then the costs should be augmented, whereas if they are already exploited, then new money to cover the real costs is needed.

4. **Assessment of Value** [51.3%]

   Concrete values given in a spreadsheet have to be interpreted by the reader as well in order to make a judgement of the data itself, where this *assessment of the value* is a trigger for putting the assessment of purpose to work. For instance, the size of the cover ratio number itself tells the informed reader whether the project is successful from a financial standpoint. If the cover is close to 100%, "everything is fine" would be one natural assessment of its value.

5. **Formula** [23.1%]

   With a given formula for a value in a spreadsheet's cell the reader knows exactly how the value was computed, so that she can verify her understanding of its intention against the author's. Note that a lot of errors in spreadsheets result from this distinction. In our experiment, if a value of a cell was calculated with a formula explicitly given in the spreadsheet, then the expert explained the dependency of the items in the formula, but restricted from just reading the formula aloud. In particular, he pointed to the respective cells and tried to convey the notion of the formula by visualizing their dependency, not so much what the dependency was about.

6. **Provenance** [43.6%]

   The *provenance* of data in a cell describes how the value of this data point was obtained, e.g. by direct measurement, by computation from other values via a spreadsheet formula, or by import from another source; see [MGM^{+}08] for a general discussion of provenance. In our interviews — as many of the data of the concrete spreadsheet were simply an output of the underlying controlling data base — the provenance explanations mostly referred to the specific data base where the data comes from. But when the formula for a value was computed, but not within Excel, the expert tried to give the formula as provenance information, e.g. in the case of the cover ratio. This knowledge was often very difficult to retrieve afterwards for the creation of the semantic document.

7. **History** [15.4%]
   The *history*, i.e., the creation process of a spreadsheet over time, often is important to understand its layout that might be inconsistent with its intention. For instance, if an organizational change occurs that alleviates the controlling process and makes certain information fragments superfluous, then those fragments will still be shown in the transition phase and beyond, even though their entropy is now 100% in the most of cases.

These seven explanation types were distilled from the recorded set of 110 explanations. The percentages given can function as a *relevance ranking* done by the expert with respect to the importance of explanation types for providing help.

Figure 4 portrays the distribution of occurrences according to each type. The "Wizard of Oz" experiment interpretation suggests that Figure 4 showcases the user requirements for SACHS as a user assistance system (see also [NW06]).

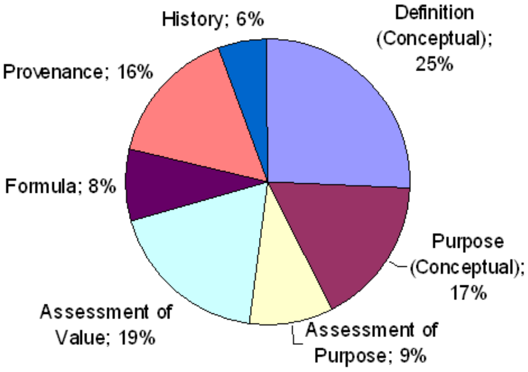


Figure 4: Help Needed — But Where?

In particular, we can now *evaluate the SACHS system* with respect to this figure. Unsurprisingly, Definition explanations were the most frequent ones. Indeed, the SACHS system addresses this explanation type either with the theory graph-based explanation interface in Figure 3 or the direct help text generator shown in Figure 2. But the next two types are not covered in the SACHS system, even though it can be argued that the ontology-based SACHS architecture is well-suited to cope with Purpose explanations — indeed, some of the purpose-level explanations have erroneously found their way into SACHS definitions, where they rather should have been classified as 'axioms and theorems' (which are currently not supported by the SACHS interface). The next explanation category (Provenance; 16%) has been anticipated in the SACHS architecture (see [KK09a]) but remains unimplemented in the SACHS system. The Assessment of Purpose type is completely missing from SACHS as well as Assessment of Value. Explanations of type Formula are only rudimentarily covered in SACHS by virtue of being a plugin that inherits the formula bar from its host application Excel, which has some formula explanation functionality. Finally, the explanation type History is also not yet covered in SACHS.

To summarize the situation: Excel is able to give help for 8% of the explanations we found in the help of a human expert. The implemented SACHS system bumps this up to 33%, while the specified SACHS system can account for 50%. Even though this is certainly an improvement, it leaves much more to be desired than we anticipated. In particular, we draw the conclusion that background knowledge that 'only' contains a domain ontology is simply not enough.

We will try to remedy parts of this in the remainder of this paper. In particular, we take up the problem of Assessment of Value explanations. On the one hand, it is ranked second in the list of explanation types with a stunningly high percentage of 51.3%, which can be interpreted as the second-best type of explanations from the point of view of our expert. On the other hand, the very nice thing about assessment for computational data is that we can hope for a formalization of its assessment in the form of formulas, which can be evaluated by the spreadsheet player in turn.

4 Modelling Assessment

A first-hand approach of complementing spreadsheets with assessment knowledge could be the inclusion of Assessment of Value information into the definition text itself. In the concrete SACHS ontology we felt that we had no other choice in order to convey as much knowledge as possible, it is ontologically speaking a very impure approach (hence wrong) as such judgements do not solely depend on the concept itself. For instance, they also depend on the respective Community of Practice: At one institution e.g. a cover ratio of 95% might be judged as necessary, at another 100% (or more) might be expected.

So before we address the question of how to model assessment, first we have to take a closer look at assessment itself: What is it about? Assessments consist of value judgements passed on situations modeled by (parts of) spreadsheets. As such, we claim that assessments are deeply in the semantic realm. To strengthen our intuition, let us consider some examples; we will use a slightly varied version of the simple spreadsheet document in Figure 1, which we have already used in [KK09a; KK09c] for this. The following can be considered typical assessment statements:

I) "*Row 6 looks good.*"
II) "*The revenues look good.*"
III) "*I like this* [points to cell [E17]] *but that* [points to cell [F17]] *is a disaster.*"
IV) "*I like the profit in 1987 but of course not that in 1988.*"
V) "*Upper Management will be happy about the left-over funds in [nn] that they can now use elsewhere, but the PI of the project will be angry that he got less work out of the project than expected. Not to mention the funding agency; they cannot be told of this at all, because it violates their subsistence policy.*"

On the surface, the first statement refers to a row in the spreadsheet, but if we look closer, then we see that this cannot really be the case, since if we shift the whole spreadsheet by one row, then we have to readjust the assessment. So it has to be about the intended meaning of row 6, i.e., the development of revenues over the years. Indeed we can paraphrase I with II — another clue that the assessments are really about situations modeled by a functional block in the spreadsheet. But assessments are not restricted to functional blocks as statements III and IV only refer to individual cells. Note again that the statements are not about the numbers 0.992 and -0.449 (numbers in themselves are not good or bad, they just are). Here the assessment seems to be intensional, i.e., about the intension "the profit in 1987/8" rather than the extension. Another way to view this is that the latter two assessments are about the argument/value pairs $\langle 1987, 0.992\rangle$ and $\langle 1988, -0.449\rangle$. We will make this view the basis of our treatment of assessment in SACHS: We extend the background ontology by a set of assessment theories that judge the intended functions in the functional blocks of the spreadsheet on their functional properties.

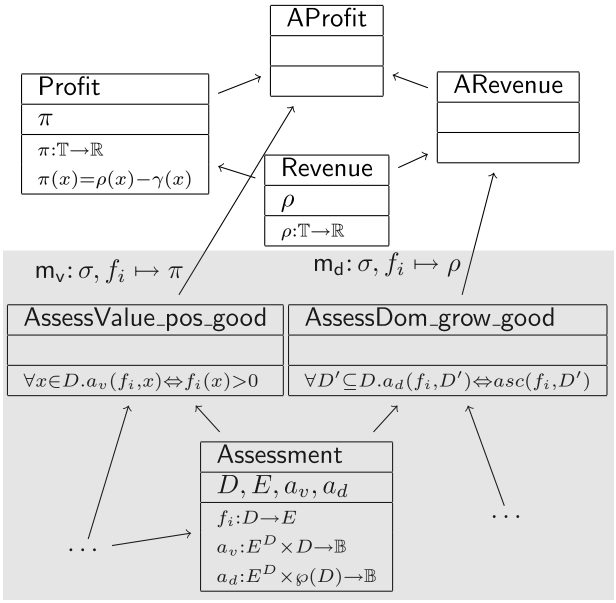
4.1 Assessment via Theories and Morphisms

Consider the partial theory graph in Figure 5, which we will use to account for the assessments in the examples I to IV above. The figure shows the theories Revenue and Profit which are part of the background knowledge, the **assessed theories** ARevenue and AProfit, and the **assessment theories** (set in the gray part) that will cover assessment itself.

The theory Assessment provides three concepts: a generic function f_i (used as a placeholder for the intended function of the functional block we are assessing), a function a_v for assessing whether a value in a cell is 'good', and finally a function a_d for assessing whether a function is 'good' over a subdomain. This generic theory — note that this does not provide any judgements yet, only the functions to judge — is then refined into concrete assessment theories by adding axioms that elaborate the judgement functions a_v and a_d, which are then used to provide concrete judgement functions to the assessed theories, via interpreting theory morphisms. The theory AssessValue_pos_good restricts the interpretation of a_v so that it assesses the function f_i as 'good' on an argument x, iff $f_i(x)$ is positive, and the theory AssessDom_grow_good restricts the interpretation of a_d to a function asc to evaluate f_i as 'good' on a subdomain $D' \subseteq D$, iff f_i is increasing on D'. Note that these assessments are still on the 'generic function' f_i over a 'generic domain' D with a 'generic range' in E. These are made concrete by the theory morphisms $\mathsf{m_v}$ and $\mathsf{m_d}$ that map these concrete sets and functions into the assessed theories, thereby applying the judgemental axioms in the assessment theories in the assessed theories.

Of course theories AssessValue_pos_good and AssessDom_grow_good are just chosen to model the examples from the start of this section. A realistic formalization of assessment, would provide a large tool-chest of theories describing the "shape" of the function f_i for knowledge engineers to choose from. With this, providing a judgement about a value becomes as simple as choosing a cell and an assessment theory: the cell determines the intended function, with its domain and range and thus the mapping of the theory morphism. Thus the assessed theory can be constructed automatically by the SACHS system.

In our example we have restricted ourselves to unary functions, but of course it is very simple to provide assessment theories for any arity that occurs in practice. Moreover, we have only used assessment theories


AProfit
Profit
π
$\pi{:}\mathbb{T}\to\mathbb{R}$
$\pi(x)=\rho(x)-\gamma(x)$
ARevenue
Revenue
ρ
$\rho{:}\mathbb{T}\to\mathbb{R}$
$\mathsf{m_v}\colon \sigma, f_i \mapsto \pi$
$\mathsf{m_d}\colon \sigma, f_i \mapsto \rho$
AssessValue_pos_good
$\forall x{\in}D.a_v(f_i,x){\Leftrightarrow}f_i(x){>}0$
AssessDom_grow_good
$\forall D'{\subseteq}D.a_d(f_i,D'){\Leftrightarrow}asc(f_i,D')$
Assessment
D, E, a_v, a_d
$f_i{:}D\to E$
$a_v{:}E^D\times D\to\mathbb{B}$
$a_d{:}E^D\times\wp(D)\to\mathbb{B}$
...
...

where $\mathbb{B}$ is the set of Boolean values, $\mathbb{R}$ is the set of real numbers, and $\mathbb{T}$ the set of time intervals (over which profits are measured). Furthermore, $\sigma := \{D \mapsto \mathbb{T}, E \mapsto \mathbb{R}\}$

Figure 5: A Partial Assessment Graph for Profits

that only refer to inherent properties of the intended functions (e.g. being monotonically increasing), but many real-world assessments are context-dependent. E.g. one might want the profit of a German Company to grow more rapidly than the DAX. This is where the knowledge-based approach we are proposing really starts to shine: we just add an assessment theory with an axiom

$$\forall t.a_v(f_i, t) \Leftrightarrow \frac{f_i(t)}{f_i(p(t))} > \frac{\mathrm{DAX}(t)}{\mathrm{DAX}(p(t))}$$

where $p(t)$ is the predecessor time interval of t.

4.2 Multi-Context Assessments and Framing

Note that the assessments above are "author assessments", since they are supposedly entered into the background ontology by the spreadsheet author. But the author's assessment is not the only relevant one for the user to know: In Example V we have a single explanation that refers to three different assessments that differ along the role of the "assessor". Multiple assessment contexts can be accommodated in our proposed model — any user of the system can enter assessments. These user assessments can even be stored in a private extension to the background ontology, if the user does not have write access to the system-provided one. In fact we can enable multi-context assessment by just providing the a_v and a_d functions with another argument that determines a fitting user or Community of Practice (see [KK06] for an introduction to Communities of Practice and their reification in the background knowledge). This will generally get us into the situation in Figure 6, where we have an assessment of profits by the author — in theory AAssessProfit — and one by the user — UAssessProfit (we have abstracted from the internal structure of the theories). The dashed ar-row is the (functional) interpretation that maps the functional block to the author-assessed theory.

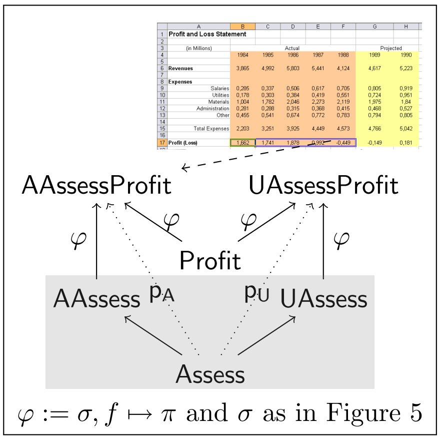
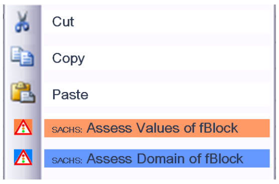


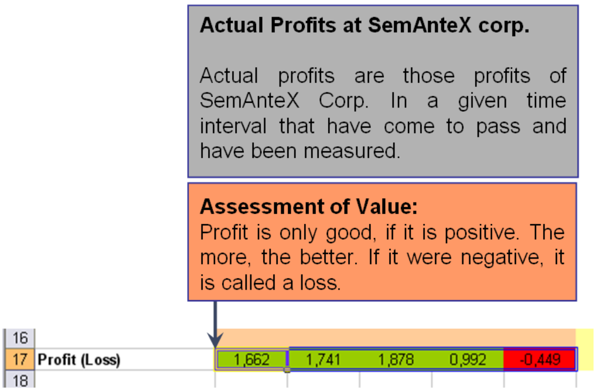
Figure 6: Multi-Context Assessment

In the framing-based user interface put forward in [KK09c] we use theory morphisms as framings and provide frame-based exploration of variants. In this example the canonical frame (the identity morphism from AAssessProfit to itself) can be generalized to the frame p_A with source theory Assess, which spans a frame variant space that includes the frame p_U and thus the user assessment, which the user can choose to explore this assessment. Needless to say, this works for any number of assessments (private or public).

5 The Envisioned Assessment Extension in SACHS


Cut
Copy
Paste
SACHS: Assess Values of fBlock
SACHS: Assess Domain of fBlock

We will now show how assessments can be made useful for the user. As the assessments are bound to (the intended function of) a functional block, we extend the context menu with entries for all assessment functions. On the left we assume a right mouse click on the cell [B17] to show the context menu with the two assessment functions a_v and a_d.



Actual Profits at SemAnteX corp.

Actual profits are those profits of SemAnteX Corp. In a given time interval that have come to pass and have been measured.

Assessment of Value:
Profit is only good, if it is positive. The more, the better. If it were negative, it is called a loss.

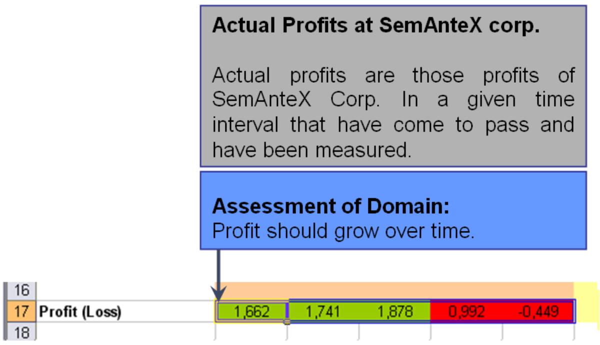
16						
17	Profit (Loss)	1,662	1,741	1,878	0,992	-0,449
18						

Figure 7: Assess the *Values* in the Functional Block

When the "Assess Values of fBlock" entry is selected SACHS is put into a special "assessment mode", which brings assessment information to the user's attention. In the background the SACHS system determines the version of the a_v axiom inherited by the AProfit, translates it into an Excel formula, and evaluates it to obtain the judgements.

Here the axiom is $\forall t.a_v(\pi, t) \Leftrightarrow \pi(t) > 0$, and it is evaluated on all cells in the functional block, resulting in the values t, t, t, t, f, which SACHS color-codes as shown in Figure 7 to warn the user of any cells that get a negative judgement.

At the same time, the assessment mode extends the explanatory labels by explanations texts from the background ontology. Selecting the menu element "Assess Domain of fBlock" gives the result in Figure 8



Actual Profits at SemAnteX corp.

Actual profits are those profits of SemAnteX Corp. In a given time interval that have come to pass and have been measured.

Assessment of Domain:
Profit should grow over time.

16						
17	Profit (Loss)	1,662	1,741	1,878	0,992	-0,449
18						

Figure 8: Assess the *Domain* in the Functional Block

But as the assessments are synchronized with the assessed theories in the background theory graph, we can also analyze the assessments for possible causes. Recall that profits are defined as the difference between revenues and expenses, it makes sense to trace assessments through the dependency graph provided by the SACHS system for understanding the definitional structure of the spreadsheet concepts. Note that this analysis is anchored to the cell: Figure 9 shows the definitional graph for the negatively assessed cell [F17] for the profits in the year 1988. Here the revenues are also negatively assessed (color-coded red in the definitional graph), so the problem might be with the revenues.

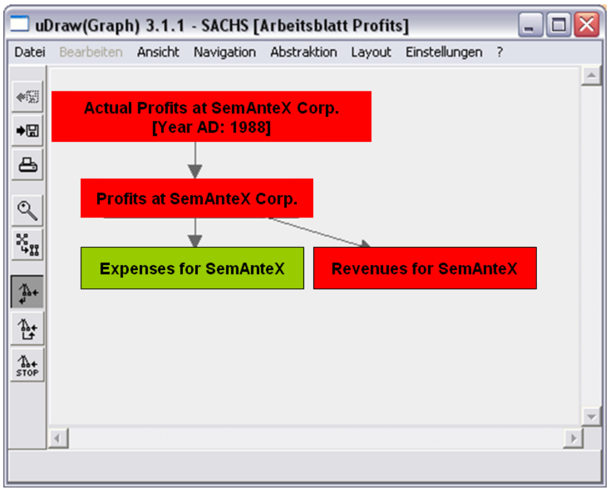

uDraw(Graph) 3.1.1 - SACHS [Arbeitsblatt Profits]
Datei
Bearbeiten
Ansicht
Navigation
Abstraktion
Layout
Einstellungen
?
Actual Profits at SemAnteX Corp. [Year AD: 1988]
Profits at SemAnteX Corp.
Expenses for SemAnteX
Revenues for SemAnteX
STOP

Figure 9: Assess the *Values* in the Dependency Graph

Note as well that this graph cannot be used for a causal analysis, as the arrows here still definitional dependency relations. We conjecture that causal analysis knowledge can transparently be included in the background ontology and can be made effective for the user in a similar interface. But we leave this for further research.

6 Conclusion and Further Work

In this paper we have reported an evaluation of the SACHS system, a semantic help system for a financial controlling system, via a (post-facto) "Wizard of Oz" experiment. The immediate results of this are twofold. The experiment basically validates the Semantic Illustration approach implemented in the SACHS system: The availability of explicitly represented background knowledge resulted in a dramatic increase of the explanations that could be delivered by the help system. But the experiment also revealed that significant categories of explanations are still systematically missing from the current setup, severely limiting the usefulness of the system. We have tried to extend the background ontology with a model of assessment to see whether the Semantic Illustration paradigm is sufficiently flexible to handle assessment.

The proposed model shows that this is indeed the case, but still has various limitations. For instance, the need to pollute the background ontology with one new theory per assessment theory and assessed theory seems somewhat unnatural and intractable even though the theories are largely empty. Also, we lack a convincing mechanism for coordinating the exploration of assessment variants: In our example in Figure 1, if we change the assessment of a profit value, we would like to change that of the respective revenue cell to a corresponding assessment.

Finally, we have only talked about Assessment of Value explanations in this paper. It seems that we can model Purpose and Assessment of Purpose explanations with a similar construction as the one proposed in Section 4: We start out with a base assessment theory which provides an assessment function like a_v, which acts on a generic intended function f_i of the functional block in question, but instead of mapping into Boolean values, it maps into a set of purposes and tasks formalized in a "task ontology" by which we would extend the background ontology. This might also make it possible to generate explanations for assessments in SACHS.

This parallelism highlights an interesting feature of the assessment model that we want to study more closely in the future. Generally, when we talk about interacting with knowledge-based systems, we have to distinguish knowledge about the system itself from knowledge structures about the domain the system addresses. We consider the first kind of knowledge as part of the *system ontology* and the second kind part of the *domain ontology*. In this sense, the assessment theories in general and in particular the function a_v provided by the theory Assessment in Figure 1 belong to the SACHS system ontology, since they have a counterpart in the implementation of the SACHS system (see Section 5), while the assessed theories clearly belong into the domain ontology. Thus, our assessment model is a very good example of the interplay of system and domain ontologies for interaction with complex applications; we conjecture that this situation will be characteristic for interaction with systems along the Semantic Illustration paradigm.

But there is also another avenue for further research: We have not made full use of the data from the "Wizard of Oz" experiment in Section 3. In Figure 10 we compute the correlations between the explanation types. The co-occurrences seem particularly interesting: as Definition is the dominating type, then the others occur relatively infrequently (from 17.9% to 50%) in the first group and the bar for Definition is relatively constant in the other clusters. The only exception to this is in the Assessment of Purpose cluster, where the co-occurrence is unusually high. Another interesting observation is that for all explanation types the co-occurrence with the Definition level is highest — except for the Purpose level. Here, the Assessment of Value statements appear more frequently than the ones of type Definition.

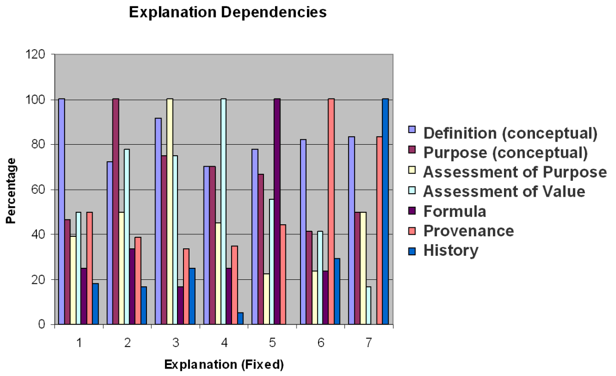


Figure 10: Explanation Dependencies

It seems that the distribution in Figure 10 might tell us something about aggregation of explanation types in help systems. To make progress on this we might try to ask: *"Given an explanation on some level, then what else knowledge is needed or useful (according to an expert)?"*. In the absence of a criterion for differentiating between necessary knowledge and voluntarily supplied knowledge in our experiment, we might take the fact that a co-occurrence above 60% seems to be an obvious critical amplitude in this tabulation as an indicator that two explanation types are 'needed or useful' for each other.

We plan to study these relationships further; if these can be corroborated in other studies and other spreadsheet-based applications, then we will fine-tune our text aggregation algorithm for the dependency graph interface in Figure 3 to volunteer the experimentally correlated explanation types.

Finally, we observe that the Semantic Illustration paradigm is neither restricted to the system Excel nor to the financial controlling domain. Unfortunately, the discussion and its consequences are beyond the scope of this paper, but was carried out in [KK09b] for user assistance systems.

Acknowledgements We like to thank all SACHS project members for their support, particularly Achim Mahnke for recording the interviews.

References

[CDSCW09] Jacques Carette, Lucas Dixon, Claudio Sacerdoti Coen, and Stephen M. Watt, editors. *MKM/Calculemus 2009 Proceedings*, number 5625 in LNAI. Springer Verlag, 2009.

[KK06] Andrea Kohlhase and Michael Kohlhase. Communities of Practice in MKM: An Extensional Model. In Jon Borwein and

William M. Farmer, editors, *Mathematical Knowledge Management, MKM'06*, number 4108 in LNAI, pages 179–193. Springer Verlag, 2006. Available from: `http://kwarc.info/kohlhase/papers/mkm06cp.pdf`.

[KK09a] Andrea Kohlhase and Michael Kohlhase. Compensating the computational bias of spreadsheets with MKM techniques. In Carette et al. [CDSCW09], pages 357–372. Available from: `http://kwarc.info/kohlhase/papers/mkm09-sachs.pdf`.

[KK09b] Andrea Kohlhase and Michael Kohlhase. Semantic transparency in user assistance systems. Submitted to SIGDOC 2009, 2009.

[KK09c] Andrea Kohlhase and Michael Kohlhase. Spreadsheet interaction with frames: Exploring a mathematical practice. In Carette et al. [CDSCW09], pages 341–256. Available from: `http://kwarc.info/kohlhase/papers/mkm09-framing.pdf`.

[KLSS09] Michael Kohlhase, Johannes Lemburg, Lutz Schröder, and Ewaryst Schulz. Formal management of CAD/CAM processes. In *16th International Symposium on Formal Methods (FM 2009*, 2009. accepted. Available from: `http://kwarc.info/kohlhase/submit/fm09.pdf`.

[Koh07] Andrea Kohlhase. Semantic PowerPoint: Content and semantic technology for educational added-value services in MS PowerPoint. In Craig Montgomerie and Jane Seale, editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007*, pages 3576–3583, Vancouver, Canada, June 2007. AACE. Available from: `http://go.editlib.org/p/25890`.

[MGM$^+$08] Luc Moreau, Paul Groth, Simon Miles, Javier Vazquez, John Ibbotson, Sheng Jiang, Steve Munroe, Omer Rana, Andreas Schreiber, Victor Tan, and Laszlo Varga. The provenance of electronic data. *Communications of the ACM*, 51(4):52–58, 2008. `doi:http://doi.acm.org/10.1145/1330311.1330323`.

[Mic] Microsoft. MS Excel. Online (`http://office.microsoft.com/en-us/excel/default.aspx`). Accessed on 2009-07-27.

[NW06] David G. Novick and Karen Ward. What users say they want in documentation. In *SIGDOC'06 Conference Proceedings*, pages 84–91. ACM, 2006.

[SGK$^+$06] Leo Sauermann, Gunnar Aastrand Grimnes, Malte Kiesel, Christiaan Fluit, Heiko Maus, Dominik Heim, Danish Nadeem, Benjamin Horak, and Andreas Dengel. Semantic desktop 2.0: The Gnowsis experience. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, *5th International Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, pages 887–900. Springer, 2006.

[Tag09] Thomas Tague. The big picture - how semantic technologies introduce a new paradigm for interaction. Invited talk at the Semantic Technology Conference, 2009. Available from: `http://www.semantic-conference.com/session/2120/`.

[Wik09] Wikipedia. Wizard of oz experiment — wikipedia, the free encyclopedia, 2009. [Online; accessed 20-May-2009]. Available from: `http://en.wikipedia.org/w/index.php?title=Wizard_of_Oz_experiment&oldid=291146893`.

[Win06] Terry Winograd. The spreadsheet. In Terry Winograd, John Bennett, Laura de Young, and Bradley Hartfield, editors, *Bringing Design to Software*, pages 228–231. Addison-Wesley, 1996 (2006).

A Mathematical Approach to Ontology Authoring and Documentation

Christoph Lange and **Michael Kohlhase**
Computer Science
Jacobs University Bremen
D-28759 Bremen, Germany
{ch.lange,m.kohlhase}@jacobs-university.de

The Semantic Web ontology languages RDFS and OWL lack practical documentation support. OWL 2 has only partly improved on that – still, ontologies cannot be annotated at all granularity levels. The languages are deliberately limited in their expressivity, but, often, one would like to express complex facts formally, too. Ways of modelling a distinction of inferred from defined facts are non-standard and not yet widely used. Finally, little work has been done on context-sensitive presentation of ontology documentation to different target audiences.

We apply to Semantic Web ontologies what we have earlier solved for mathematical knowledge representation with the OMDoc language. We model ontologies as mathematical theories, which can contain diverse types of statements, such as symbol declarations, definitions, axioms, theorems, proofs, and examples. Statements contain formulae composed of symbols; every symbol has a URI and thus corresponds to a Semantic Web resource. Additionally, we benefit from OMDoc's document infrastructure, which supports alternative human-readable notations for symbols, parallel occurrences of formal representation and informal description, and a sectional structure. Having reimplemented the Semantic Web ontology languages RDFS and OWL as theories in OMDoc, we can now author [documentations of] concrete ontologies as theories importing the former. From the same source, we can generate both a human-readable XHTML+RDFa+MathML document prepared for interactive browsing and a formal RDFS or OWL ontology in its native RDF representation, suitable for existing ontology-based tools like reasoners. Our semantic wiki SWiM serves as an integrated tool for editing and browsing OMDoc ontologies and their documentation.

We evaluated our approach by reimplementing the FOAF ontology and specification in one coherent OMDoc document and observed that 1. Imports of other ontologies could be documented. 2. For all information that was only given as a comment in the source code of the FOAF a proper OMDoc counterpart existed. 3. OMDoc allowed for documenting inferred statements properly. 4. We were able to formally mode one axiom that exceeded the expressivity of OWL, plus several facts that had only been given as an informal advice in FOAF.

Future work will focus on documenting modular ontologies and improving the editing support.

Resubmission of a paper published in: J. Carette, L. Dixon, C. Sacerdoti Coen, S. M. Watt, *Intelligent Computer Mathematics*, 8th Int. Conf. MKM 2009, Springer LNCS, Vol. 5625, pp. 389–404. A preprint can be found at https://svn.omdoc.org/repos/omdoc/trunk/doc/blue/foaf/mkm09.pdf.

An Ontology-Based Autonomic System for Improving Data Warehouses by Cache Allocation Management

Vlad Nicolicin Georgescu and Vincent Benatier
SP2 Solutions, www.sp2.fr, vladgeorgescun@sp2.fr
Remi Lehn and Henri Briand
LINA CNRS 6241 - COD Team – Ecole Polytechnique de l'Université de Nantes

Abstract

With the increase in the amount and complexity of information, data warehouse performance has become a constant issue, especially for decision support systems. As a consequence, decision experts are faced with the management of all this information, and thus realize that special techniques are required to keep good performances. This paper proposes an approach to data warehouse systems improvement based on Autonomic Computing. The idea is that by rendering certain tasks autonomic, such as the configuration of cache memory allocations between groups of data warehouses, we ensure a better data warehouse management at a lower cost and save substantial decision experts' time that can be used on higher level decision tasks.

1 Introduction

Decision Support Systems are defined as computerized systems whose main goal is to analyze a series of facts and give various propositions for actions regarding the facts involved [Druzdel and Flynn (1999)]. The process of decision making in enterprises based on such systems is also known as Business Intelligence. This concept is very well applied by large enterprises. Via this process, they specifically focus on their data warehouse efforts. The problem is that data warehouses usually become fast very large and complex, thus their performances become rapidly an issue. This is why between 70 and 90% [Frolick and Lindsey (2003)] of the enterprises consider that their data warehouse efforts is inefficient, as in many cases, the large amount of data involved becomes unusable. In many of these cases, the cause is bad management or costs that are too high to sustain.

One of the main problems that lead to this is common resource sharing between data warehouses. The resources are usually limited either by financial costs or by architectural considerations. Consider the following real example, to emphasize the problematic. An enterprise has a special server for its data warehouses. In total, a group of 50 data warehouses that share the same RAM memory is deployed on this server. Each of the data warehouse requires at least 20 GB of RAM to have good performances (i.e. the query average time is under a second). So there is a need for at least 1TB of RAM (ignoring all other RAM requirements of the server). First, the costs of having 1TB of RAM on server are financially high (~ 40000 EUR[1]). Second, if the enterprise is ready to cover these costs, suppose the server has an architecture that enables a maximum of 16GB to be installed. Also the migration of some data warehouses on another server would be too expensive and too complicated. An option is to compromise, asking each time an expert to re-configure the memory allocation for each of the data warehouse. In a short time after this is done, with the evolution of the data warehouses' size or if new data warehouses are added or some become obsolete, the problem reappears and the same action must be taken, over and over again.

Based on the example above we can intuitively see a simple solution: enable autonomic tasks that reconfigure the memory allocations, instead of asking a human expert each time to intervene (human resources are the most expensive, and not always provide the optimal results). This is easy to be said but it is hard to formalize, due to two main issues.

First, how to formally represent the group of data warehouses along with the knowledge involved in the decisions and actions of the expert? To do this, we differentiated three main types of information that needs to be formalized: a) architectural information (how a group of data warehouse is organized, the number of groups, how are they linked, etc.); b) configuration and performance information (how much memory each data warehouse needs, what performance is achieved with this allocation, etc.); and c) experience information that represents best practices and advices for the memory allocations (coming from editor documents, human experience, etc.). We present in this paper a formalization of the three types into a unified knowledge base, using ontologies [Gruber (1992)] and ontology based rules [Stojanovic et al.(2004)].

Second, having the information formalized, we need an organized form of rendering the autonomic process. To this end, IBM proposes a solution called Autonomic Computing [IBM (2001)]. It consists in the division of the actions that are taken when trying to provide autonomy to a process, corresponding to objective-specific phases and states. Autonomic concepts can be integrated in hierarchical organized systems, so each higher level aggregates what has been done to its sub levels. There are numerous autonomic computing based works that relate especially to problem resolution [Manoel et al.(2005)] or system administration [Barret et al.(2004)]. On the other hand, little has been done on data warehouse improvement.

[1] http://www.materiel.net/ctl/Serveurs/

So, we propose to use autonomic computing on the unified formalized knowledge base. Specifically, we treat a common configuration problem: cache memory allocation for a group of data warehouses (that share the same amount of common available RAM memory). The objective is to reach a better performance (in terms of query response times when extracting data from the data warehouse) with lower costs. The implication is that by increasing the amount of cached data, there are better chances that a request hits the cache; the response times in order to extract the data decrease, which translates in better performances. But, the whole amount of data obviously can't be put in the cache, and then we need a way to automatically determine and adjust the cache parameters.

Section 2 presents a view of data warehouse management through caches in the context of decision support systems. It presents the information that needs to be manipulated and how the division of this information can lead to a unified knowledge base representation. Section 3 presents how autonomic computing is used with managing data warehouse through caches. It presents how the knowledge base is integrated to permit autonomic tasks. It equally proposes two heuristics for cache allocation, based on the problematic described. Section 4 shows how we integrate the elements together using ontologies for the knowledge base representation and ontology based rules for the autonomic process. We provide some results obtained with our approach. In the conclusion we sum the work presented giving future directions and hoping that our work could help enterprises with their data warehouse efforts.

2 Data Warehouse and Cache Allocations

First of all, when speaking of data warehouse we usually make reference to a definition as a repository of an organization's electronically stored data and is designed to facilitate reporting and analysis [Inmon (2005)]. Managing a data warehouse includes the process of analyzing, extracting, transforming and loading data and metadata. As decisional experts, we know that once data warehouses are put in place, enterprises then base their decisions on the data that is stored within them. So a good organization in start and a good performance in time are the requirements of data warehousing.

We do not put into question the initial organization. We observed that in time data warehouse performances are constantly degrading up to a point where the system is no longer usable. One aspect of data warehouse performance is strongly related to the operation of data extraction which in turn depends on the query response times on the data. Obviously, the larger a data warehouse is, the more information it contains so we expect to have higher response times. Considering that some information is often more demanded than other, data warehouse management systems offer the possibility of keeping frequently accessed data in cache memories with the hypothesis that fetching data from the cache is greatly faster than fetching them from the persistent storage media. The problem occurs when confronted with groups of data warehouse on the same machine that share the same amount of memory. In decision support systems, such groups contain data of up to several thousand gigabytes. They cannot be all put into the cache, so solutions are required.

Although the problematic of performance improvement in data warehouses throughout caches is debated [Malik et al.(2008)], [Saharia and Babad (2000)] the issue is always addressed either through the physical design or the design of algorithms to determine which information is likely to be stored in cache memories. These solutions apply well when we focus on a single data warehouse.

So, what actually happens in enterprises is that the initial cache allocations remain the same throughout time. Whereas the quantity of data in the data warehouse increases, some of them are no longer used; there are new data warehouses that are constructed etc. Therefore there is a need for a dynamic system.

The first aspect of the system we propose and that we approach is knowledge formalization. In the example presented in the introduction, the expert in order to reallocate the memory makes use of several types of information. We propose to divide this information into three main types, detailing and exemplifying based on the Hyperion Essbase[2] business intelligence solution.

Architectural information corresponds to the organization of the groups of data warehouse. Figure 1 shows an example of a possible organization.

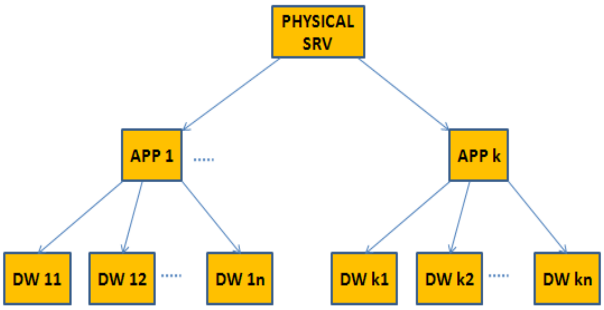


Figure 1 - Architectural organization for groups of data warehouse

Based on a decision support system simple organization, we can distinguish on top of the tree a Physical Server as the actual machine. Underneath, there are a number of Applications installed that share the RAM memory available on the server. And, in turn each application contains one or more data warehouses (Essbase cubes or bases), sharing the same memory. Each application is seen as a group of data warehouses, and then memory reallocation is done within each application.

Configuration and performance information contains all the indicators that reflect the actual characteristics and configuration of the data warehouses and the performances obtained with this configuration. For the characteristics and configuration we refer to the Essbase cubes. There are many characteristics, but for our example we take into consideration the following indicators:

- The size of each data warehouse represented by: the Index File size and the Data File size. This corresponds to the actual size that each data warehouse is occupying on a hard disk. A value of tens of GB for the two together is a frequently met characteristic.
- The values of three types of caches: Index, DataFile and Data Cache. Corresponding to the sizes presented before, they represent a percentage of the actual data files that can be kept in cache. Ideally, we should have the total of index file size in the index

[2] http://download.oracle.com/docs/cd/E10530_01/doc/epm.931/html_esb_dbag/frameset.htm?dstcache.htm

cache, and the total of data file size in the data and data file cache.

For the performance aspect, there are many indicators to take into consideration such as: query response times, calculation times, aggregation operation times, etc. We chose the query response time as a performance indicator as this is a frequently used measure [Saharia and Babad (2000)] of the system performance, and, it directly reflects the quality of the user experience in the decision system. It represents the time needed to extract data through a query from the data warehouse.

Experience and best practices information represent a more delicate subject in comparison with the first two information types. The main reason is that it comes from several different sources. Therefore the challenge is how to combine these sources into a single unified knowledge base. For instance, how to combine practices taken from an Essbase support document with practices that are part of the human experience and that are only known by the expert. We present here the formalization aspect, that is revised and validated by a human expert. In order to formalize the experience and best practices, we have found a completely different approach to knowledge representation, which is the rule based representation. Basically, we translate the pieces of advice and best practices into Event Condition Action (ECA [Huebscher and McCann (2008)]) rules. Such rules are often associated with business intelligence practices, and integrating different rules at different timelines (via the autonomic aspect) proved to be a good choice for our proposition. ECA rules have certain drawbacks, such as it is hard to prove the coherence and the no contradiction. But for the rules in our system, this aspect is not currently an issue.

3 Driving the data warehouse – Autonomic Computing

Once the principal knowledge types are well separated and formalized, they have to be 'put to life'. We refer of course at the second aspect of the improvement system: rendering it autonomic. Autonomic systems have been present within our everyday lives. A very intuitive example of an autonomic system that manages itself is the human body. Reflexes like breathing, digestion, heart pulsation etc. are part of the autonomy the human body provides (we don't control these we just know they are continually present and moreover they function). Starting from this idea, the first approaches were especially towards self-healing systems, the survey of [Ghosh et al.(2007)] summing up this evolution. And, as expected the concept developed, and in 2001, IBM proposed a formalization of the self-x factor by introducing the notion of Autonomic Computing (AC) [IBM (2001)]. Most of the IT organizations spend a lot of time reacting to problems that occur at the IT infrastructure component level. This prevents them from focusing on monitoring their systems and from being able to predict and prevent problems before end users are impacted [IBM (2005)]. Autonomic computing is the ability for an IT infrastructure to adapt and change in accordance with business policies and objectives. Quite simply, it is about freeing IT professionals to focus on higher–value tasks by making technology work smarter, with business rules guiding systems to be self-configuring, self-healing, self-optimizing and self-protecting [IBM (2001)].

From this to applying autonomic computing to enable improvement in IT infrastructures was just a small step. The subject proved to be of great interest to enterprises. Works have been done in this area and put into practice for improving database performance by IBM [Markl et al.(2003)], [Lightstone et al.(2002)] and Microsoft [Mateen et al.(2008)]. IBM specifications link autonomic computing with the notion of autonomic manager as the entity that coordinates the activity of the autonomic process. An autonomic manager (ACM) is an implementation that automates the self-management function and externalizes this function according to the behavior defined by management interfaces. The autonomic manager is a component that implements an intelligent control loop. For a system component to be self-managing, it must have an automated method to collect the details it needs from the system (Monitor); to analyze those details to determine if something needs to change (Analyze); to create a plan, or sequence of actions, that specifies the necessary changes (Plan); and to perform those actions (Execute) [IBM (2001)]. Similar alternatives to autonomic computing were made in real BI [Nguyen et al.(2005)] but the idea is the same: to be able to analyze and improve (in our case) a given system through a closed loop that differentiates a series of states.

We propose the usage of autonomic managers to enable data warehouse self-improvement. Figure 2 shows the transformation of the architecture from Figure 1, with the implementation of autonomic managers on each of the entities (or component of the architecture) involved.

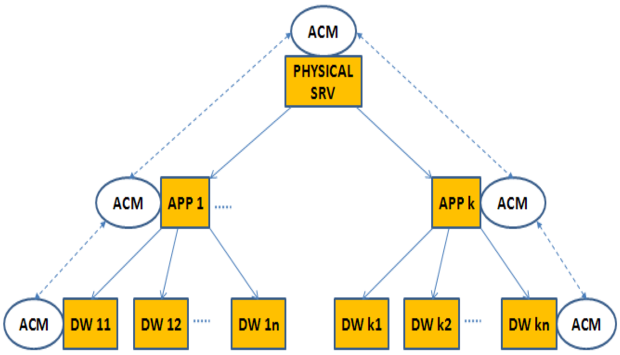


Figure 2 - Autonomic Computing Managers on each of the architectural levels

We notice that each of the entities has its own individual loop. The autonomic managers communicate only with the ones from the superior levels, and not between the same level. This way, each entity has two responsibilities: one strictly related to its individual self management and the other related to the management of its descendants. The idea is that the two can function independently of each other. For instance, consider an Application that has 2GB of RAM allocated to its data warehouses. So each data warehouse uses the allocated RAM and self-improves itself with what it has. Now suppose that at a certain point the Application receives another 1GB of RAM. If the new information is not integrated then the data warehouses continue to function with the already allocated 2GB. Once the application runs the management of its descendants, a reallocation of the memory is done also for the data warehouses. In order to simulate the two behaviors, we have elaborated two heuristics.

3.1 Data warehouse Self-improvement heuristic

This concerns only the individual loop at a data warehouse level. Its role is to describe how cache allocations vary with the query response times. The idea is the following: starting from a given maximal cache configuration we try to decrease the values of the caches and study the impact this decrease has on the data warehouse query response times. The algorithm stops when the difference between the current and the last average query response time is greater than a specified threshold. This is done independently for each data warehouse. So, we define two parameters for this heuristics:

Step - represents the amount with which each cache value is decreased. The following formula shows how a cache value modifies with step:

$$CV_1 = CV_0 - (CV_{max} - CV_0)*step$$

where CV_0 represents the old cache value, CV_1 the new calculated value and CV_{max} the maximum possible value . A frequent value of step we used in our experiments was 10% based on the recommendation of our human experts.

Delta – represents the threshold accepted for the difference between the current and the last average response time. It can be seen as the accepted impact that a cache modification has. If $(RT_1-RT_0)/RT_0 < delta$ then we accept the new cache proposition (where *RT = average response time* for the respective data warehouse). A frequent value we used for delta was 5%, based on our clients' average performance acceptance specification (i.e. for a value of x, an fluctuation in performance with 5% is accepted).

Table 1 illustrates the self-improvement heuristics with a timeline, based on the autonomic manager loop phases. At t_0 we have the initial configuration. At t_1 we have made the first cache adjustment, and validated it. At t_2, the second cache modification has an impact too great on the response time so we leave the cache value as it is.

Table 1 - Individual Data Warehouse Self-Improvement Heuristics on the autonomic manager phases

Time	AML Phase	Action
0	Monitor	step = 0.1, delta = 0.05, CV_{max}=1GB CV_0=500MB, RT_0=5s
	Analyze	N(ot)/A(vailable)
	Plan	CV_1=450MB
	Execute	Change script for DW with CV_1
1	Monitor	CV_1=450MB, RT_1=5.2s
	Analyze	$(RT_1-RT_0)/RT_0$=0.04 < delta
	Plan	CV_2=395MB
	Execute	Change script for DW with CV_2
2	Monitor	CV_2=395MB, RT_2=6s
	Analyze	$(RT_2-RT_1)/RT_1$=0.15 > delta
	Plan	CV_3=395MB
	Execute	No change for DW

3.2 Group of data warehouses cache reallocation heuristic

The first heuristics was individual data warehouse based. Each of the data warehouses was independent and each was in a state of self-improvement in time. But, taking it into consideration alone makes no sense as the performances on individual data warehouses are expected to decrease as the caches decrease. To explain how it results in an actual improvement at group level, we introduce the group of data warehouses heuristic. Its purpose is to reallocate periodically the memory that the individual data warehouse heuristics saved from the self-improvement process. And it is here where the 'catch' is: by a small sacrifice (delta) of some data warehouses, we can gain an important performance on others.

The core of the heuristic is to differentiate the non performing from the performing data warehouses in a group. The idea is the following: a data warehouse is considered performing if its average response time is below the average value of the response time for the whole group. Otherwise, it is considered as non-performing. This performance indicator can be equally made more complex by taking into account the applications priority or importance. This way scaled mixed performance indicators can be obtained and used. The specification of priorities and importance is usually part of Service License Agreements and is one of the future directions in our work.

So in this case, we take the memory from the performing data warehouse and give it to the non-performing. Relating with the architecture in Figure 1, the Application level is responsible for the implementation of this heuristic. The Application decides how to redistribute the memory between the data warehouses it concerns.

Table 2 shows this heuristics. The example is based on a group of two data warehouses that are part of the same application and share the same amount of memory for their caches.

Table 2 - Group of Data Warehouse Improvement Heuristic – Cache Evolution Exemple

Step	DW	Cache Value	Memory to allocate	Free Memory	RT
0	DW1	130 MB	140 MB	10 MB	5s
	DW2	80 MB	90 MB	10 MB	7s
1	DW1	130 MB	130 MB	0 MB	5s
	DW2	100 MB	100 MB	0 MB	6s
2	DW1	120 MB	120 MB	0 MB	5.3s
	DW2	110 MB	110 MB	0 MB	5.5s

In start at s_0 we have a given cache allocation along with the available memory for each data warehouse. At s_1 the heuristic is run the first time. It takes all the available memory from the performing data warehouse (DW1) and redistributes it to the non performing (DW2). So DW2 gains all the free memory (20MB) from s_0. As the differences in response times are still important, it goes further at s_2. Here, it takes some memory from DW1 by force, leading to a decrease in performance for DW2. But as seen, we gain an important amount of performance for DW2, and now the response times for the two data warehouses are close.

It is important to note that this heuristic is independent from the previous one, and in addition the two heuristics are mutually exclusive. This means that in the moments when this heuristic is considered, the other does nothing. This is why between the two tables we differentiate between "Time" and "Step". An example of usage is to run the individual self-improvement heuristic once each day (from Tuesday to Friday), and the group reallocation heuristic once at the beginning of each weak (Monday).

4 Combining the elements

Having the two main aspects, knowledge formalization and autonomic capabilities, the final and innovative stage in our approach is to combine them. In order to do this we base on the preliminary works presented in [[Nicolicin-Georgescu et al.(2009)]]. The solution proposed the application of ontologies and ontology based rules (describing business rules) with autonomic computing for improving average query response times in data warehouse. The concept is the same, but in this previous work we only described how can simple businesses rules can be used to improve data warehouse performance. There is no indication to how heuristics are used within the autonomic manager loop.

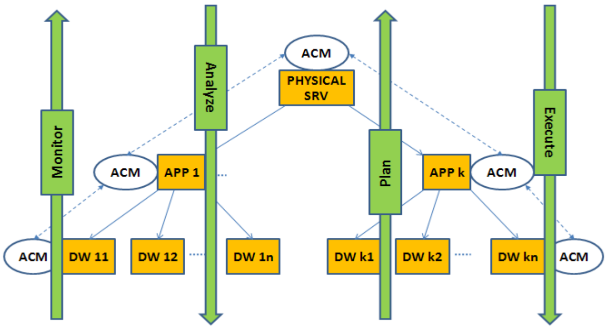
4.1 System implementation

In previous works, we were proposing a division of the knowledge in the system into static knowledge and dynamic knowledge. Based on this organization we implement the new presented elements. The means of knowledge formalization do not change, static knowledge being implemented with the help of ontologies and OWL[3] whereas the dynamic aspect is expressed with ontology based rules via the Jena Rules[4] formalization. The ontology contains over 150 concepts and 250 axioms, whereas a number of 30 rules are based on it. From what we have presented, we focus on the dynamic aspect, as it includes the two heuristics projected on the phases of the autonomic computing.

The first step in order to understand how rules are organized is to understand how the autonomic managers on the different hierarchical levels communicate. As seen the group heuristic reallocates memory and excludes the individual heuristic. In order for the autonomic managers to communicate, we propose a hierarchy of the autonomic phases, corresponding to the architectural structure. Figure 3 show how the four phases of autonomic manager are projected on the architectural levels.

We notice how the monitor phases ascends, starting from the lowest level (data warehouse). This means that first the data regarding the data warehouses are gathered, then the application, and then the physical server. Then the analysis is made top down from the physical server to the data warehouse level. Retaking the memory allocation example, first the server allocates memory between its applications, then each application allocates in turn to its data warehouses etc. Then the planning stage ascends again, the changes are planned from the analysis level starting with the data warehouses and finishing with the physical server. Last, the execution phase makes changes top down similar to the analyze phase. A change in the RAM memory is first done to the physical server, then the applications receive the new memory and then the data warehouses change their memory (now possible because the memory has already been changed at application level).

Figure 3 - Autonomic Manager phases projection on the architectural levels



We exemplify below how the system is implemented on each of the four phases.

Monitor

For the monitor phase, in order to obtain the cache values and average response times, we use SQL data bases that are filled with the help of vbscripts via the api provided by Hyperion Essbase. Then, to transform and load this knowledge in the ontology, we pass via a java program using the Jena API and a set of correspondences that links the data from the SQLdbs to ontology concepts. Table 3 shows how some parts of how a data warehouse is represented in the ontology:

Table 3 - Data Warehouse ontology representation

Subject	Predicate	Object
?dw	rdf:type	DataWarehouse
?app	rdf:type	PhysicalApplication
?dw	isChildOf	?app
?dw	hasAvgResponseTime	?avgt
?dw	hasPrevAvgResponseTime	?prevt

We can see two classes, the DataWarehouse and the PhysicalApplication. Each of these classes consist from multiple instances as OWL individuals. The ?dw is one such individual that is linked to an ?app individual by the OWL object property isChildOf . This property establishes the hierarchical relations between individuals from the different hierarchical levels. Then, there are two OWL data type properties that are linked to the ?dw and express the current and previous average response time for ?dw. The values for these properties are filled from the SQL dbs that contains to the data warehouse monitor information.

Analyze

Once this phase of monitoring and pre-processing of information is done, the system passes to analyze. We present below two rules that formalize a cache decrease.

Rule	Description
(?dw cp:hasPrevAvgResponseTime ?prevt) (?dw cp:hasAvgResponseTime ?avgt) (?dw cp:hasAlgorithm ?alg) (?alg cp:hasDelta ?delta) quotient(?t, ?avgt, ?prevt) le(?t, ?delta) -> (?dw cp:hasState cp:DecreaseCache)	Validate a cache decrease via the individual heuristic
(?dw cp:hasState cp:DecreaseCache) (?dw cp:hasIndexCacheMin ?ic_min) (?dw cp:hasIndexCache ?ic) (?dw cp:hasAlgorithm ?alg) (?alg cp:hasStep ?step) product(?p, ?ic, ?step) difference(?ic_new, ?ic, ?p) ge(?ic_new, ?ic_min) -> (?dw cp:hasIndexCache ?ic_new)	If the decrease of cache is requested, test if the new value is not under the minimal value. If not enable the new change.

The first rule test to whether the cache values for a single data warehouse can be decreased, accordingly with the individual heuristic. We have again the ?dw individual, an instance of the DataWarehouse class, with the two data type properties from Table 4. In addition we have a new object property that related the ?dw with an the individual heuristic algorithm. The rule compares the rapport between the two average times (current and previous) with the delta of the algorithm. If the rapport is lower than delta, the ?dw becomes into a new state, in which it is allowed to decrease its cache. Otherwise, nothing changes.

[3] http://www.w3.org/2007/OWL/wiki/OWL_Working_Group

[4] http://jena.sourceforge.net/inference/#rules

The second rule makes use of the results of the first rule. If the DecreaseCache state has been generated by the first rule, then it tries to see whether or not the operation is possible. Two new data type properties are introduced for the ?dw: hasIndexCacheMin and hasIndexCache, which represent the values of the minimum threshold and the current value of the cubes index cache. These values are equally filled from the monitor phase. The rule retakes the individual heuristic algorithms' parameters (step this time) and tests if by modifying the index cache with its formula, the new index cache value is greater than the minimum one. If so, it changes directly the current index cache value to the newly computed one.

Plan and Execute

The plan and execute phases are linked to each other. As the new cache values are calculated, there is a preparation of VBscripts that will be run via the program. These scripts will change the values of the caches in the Essbase cubes, according to the new values proposed by the analyze phase. At the end of the execution phase, practically the inverse monitor operation of data processing is made. The cache values are passed from the ontology to the SQLdbs and then to the modification scripts.

4.2 Experimentation and Results

For our experiments we considered the following scenario: on an existing server, we created an Essbase application with two cubes. The cubes contain in average 11 principal axes and 27 level 2 axes and the data file has an average size of 300MB. With this configuration, we carried several tests, simulating a period of 14 days (time stamps period). Each time-stamp, a series of random queries (from a given set) was executed so that activity on the application was simulated. The individual data warehouse self improvement heuristic is running each day, whereas the group heuristic is running once each 4 days. Figure 4 shows the evolution of the response times for the two data warehouses with the evolution of their total cache allocation:

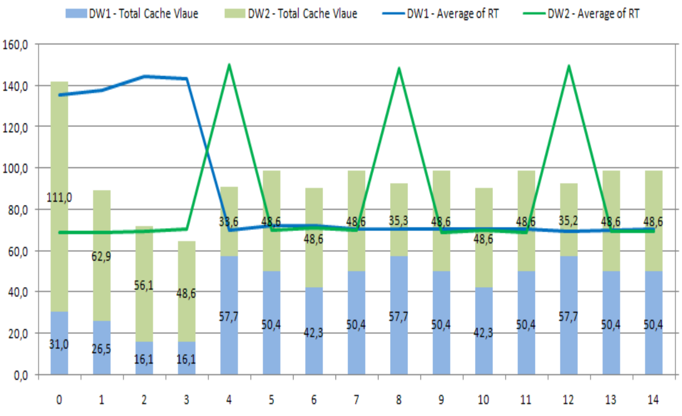


Figure 4 - Average Response Time evolution with cache allocations

Again, the objective is to obtain better average response times with lower cache values. First what we notice is that at the end of 5 days we already have a good ratio response time/cache allocation. The data warehouses improve themselves fast, and then once reaching a good point, they oscillate around this point. This oscillation is shown by the peaks on DW2 that tries each time to improve more its performances, but it can't due to the heuristic constraints in terms of delta. Their impact on the system is felt in terms of the performance drops the days where the peaks are noticed. By limiting the number of peaks (i.e. after one or two peaks the system should no longer try to optimize under the same circumstances) we avoid the risk of such drops. But, we also have to take into consideration that by limiting the number of peaks (by forcing the algorithm to stop for instance at a certain point) we risk to miss some needs of improvement due to reconfiguration aspects. The ideal would be to leave the heuristic running as usual and not to force the algorithm to stop, but not to accept the cache decreases once a certain level of performance is reached. One of the future directions and improvements is the introduction of attenuation mechanisms in the loop.

So, in numbers, at the end of the sixth day: DW1 looses very little in performances (~2%), DW2 gains substantial performances (~80%), and the total cache used by the application is decreased by ~60%. So the sacrifice of DW1 was worth from the perspective of the entire system. These results prove how an efficient way of improving the data warehouse group performances can be achieved in an autonomic manner, without the intervention of a human expert.

5 Conclusions

This article presented a way of using ontologies and autonomic computing for improving query response times in data warehouses groups by modifying cache memory allocations. It has presented this applied to the problematic of shared resource allocation in groups of data warehouses. Also, the article presented a proposition of replacing some of the human expert's work by introducing autonomic and human independent ways of managing data warehouses.

We have proposed a division and formalization of the knowledge used for configuring groups of data warehouses by using ontology and ontology based rules. Also, we have proposed an organization of this process based on the autonomic computing considerations. It is not the first attempt to combine the two [Stojanovic et al.(2004)], but the novelty is from using such techniques in the domain of decision support systems and especially in the groups of data warehouse improvement.

Our future directions are to expand the data warehouses described above so that our prototype can prove its efficiency on a larger spectrum of rules and indicators. Our purpose is to integrate the prototype presented here with more than one aspect (data warehouse cache allocations based on response times) of decision support systems. We also intend to approach the notions of Service License Agreement (SLA) and Quality of Service (QoS), by introducing the QoS as a performance indicator in the system. SLA considerations such as application priority and importance depending on utilization periods, are two aspects that are little approached and equally very important in a decision support system. Also in terms of autonomic loop control, we take into consideration the usage of mechanisms for avoiding peaks and unnecessary loop passages.

As the domain is relatively new we try to bring as much support as possible for future development in the direction of autonomic knowledge based decision support systems. We follow the changes with the new technologies and hope that our work will be useful in this expanding environment.

References

[Barret *et al.*(2004)] Rob Barret, Paul P. Maglio, Eser Kandogan, and John Bailey. Usable autonomic computing systems: the administrator's perspective. In *ICAC 2004*, 2004.

[Druzdel and Flynn (1999)]M.J. Druzdel and R.R. Flynn. *Decision Support Systems*. Encyclopedia of library and information science, 1999.

[Frolick and Lindsey (2003)] Mark N. Frolick and Keith Lindsey. Critical factors for data warehouse failure. *Business Intelligence Journal*, Vol. 8, No. 3, 2003.

[Ghosh *et al.*(2007)] Debanjan Ghosh, Raj Sharman, H. Raghav Rao, and Shambhu Upadhyaya. Self-healing systems — survey and synthesis. *Decision Support Systems 42*, Vol 42:p. 2164–2185, 2007.

[Gruber (1992)] T. Gruber. *What is an ontology?* Academic Press Pub., 1992.

[Huebscher and McCann (2008)] M.C. Huebscher and J.A. McCann. A survey on autonomic computing – degrees, models and applications. *ACM Computing Surveys*, Vol. 40, No. 3, 2008.

[IBM (2001)] Corporation IBM. *An architectural blueprint for autonomic computing*. IBMCorporation, 2001.

[IBM (2005)] Corporation IBM. Autonomic computing. powering your business for success. *International Journal of Computer Science and Network Security*, Vol.7 No.10:p. 2–4, 2005.

[Inmon (2005)] W.H. Inmon. *Building the data warehouse, fourth edition*. Wiley Publishing, 2005.

[Lightstone *et al.*(2002)] S.S. Lightstone, G. Lohman, and D. Zilio. Toward autonomic computing with db2 universal database. *ACM SIGMOD Record*, Vol. 31, Issue 3, 2002.

[Malik *et al.*(2008)]T. Malik, X. Wang, R. Burn, D. Dash, and A. Ailamaki. Automated physical design in database caching. In *ICDE Workshop*, 2008.

[Manoel *et al.*(2005)] E. Manoel, M.J. Nielsen, A. Salahshour, S. Sampath, and S. Sudarshanan. Problem determination using self-managing autonomic technology. *IBM RedBook*, pages p. 5 – 9, 2005.

[Markl *et al.*(2003)]V. Markl, G. M. Lohman, and V. Raman. Leo : An autonomic optimizer for db2. *IBM Systems Journal*, Vol. 42, No. 1, 2003.

[Mateen *et al.*(2008)] A. Mateen, B. Raza, and T. Hussain. Autonomic computing in sql server. In *7th IEEE/ACIS International Conference on Computer and Information Science*, 2008.

[Nguyen *et al.*(2005)] T. M. Nguyen, J. Schiefer, and A. Min Tjoa. Sense & response service architecture (saresa). In *DOLAP'05*, 2005.

[Nicolicin-Georgescu *et al.*(2009)] Vlad Nicolicin-Georgescu, Vincent Benatier, Remi Lehn, and Henri Briand. An ontology-based autonomic system for improving data warehouse performances. In *Knowledge-Based and Intelligent Information and Engineering Systems, 13th International Conference, KES2009*, 2009.

[Saharia and Babad (2000)]A. N. Saharia and Y.M. Babad. Enhancing data warehouse performance through query caching. *The DATA BASE Advances in Informatics Systems*, Vol 31, No.3, 2000.

[Stojanovic *et al.*(2004)] L. Stojanovic, J. Schneider, A. Maedche, S. Libischer, R. Studer, Th. Lumpp, A. Abecker, G. Breiter, and J. Dinger. The role of ontologies in autonomic computing systems. *IBM Systems Journal*, Vol. 43, No. 3:p. 598–616, 2004.

Barret *et al.*(2004)] Rob Barret, Paul P. Maglio, Eser Kandogan, and John Bailey. Usable autonomic computing systems: the administrator's perspective. In *ICAC 2004*, 2004.

[Druzdel and Flynn (1999)]M.J. Druzdel and R.R. Flynn. *Decision Support Systems*. Encyclopedia of library and information science, 1999.

[Frolick and Lindsey (2003)] Mark N. Frolick and Keith Lindsey. Critical factors for data warehouse failure. *Business Intelligence Journal*, Vol. 8, No. 3, 2003.

[Ghosh *et al.*(2007)] Debanjan Ghosh, Raj Sharman, H. Raghav Rao, and Shambhu Upadhyaya. Self-healing systems — survey and synthesis. *Decision Support Systems 42*, Vol 42:p. 2164–2185, 2007.

[Gruber (1992)] T. Gruber. *What is an ontology?* Academic Press Pub., 1992.

[Huebscher and McCann (2008)] M.C. Huebscher and J.A. McCann. A survey on autonomic computing – degrees, models and applications. *ACM Computing Surveys*, Vol. 40, No. 3, 2008.

[IBM (2001)] Corporation IBM. *An architectural blueprint for autonomic computing*. IBMCorporation, 2001.

[IBM (2005)] Corporation IBM. Autonomic computing. powering your business for success. *International Journal of Computer Science and Network Security*, Vol.7 No.10:p. 2–4, 2005.

[Inmon (2005)] W.H. Inmon. *Building the data warehouse, fourth edition*. Wiley Publishing, 2005.

[Lightstone *et al.*(2002)] S.S. Lightstone, G. Lohman, and D. Zilio. Toward autonomic computing with db2 universal database. *ACM SIGMOD Record*, Vol. 31, Issue 3, 2002.

[Malik *et al.*(2008)]T. Malik, X. Wang, R. Burn, D. Dash, and A. Ailamaki. Automated physical design in database caching. In *ICDE Workshop*, 2008.

[Manoel *et al.*(2005)] E. Manoel, M.J. Nielsen, A. Salahshour, S. Sampath, and S. Sudarshanan. Problem determination using self-managing autonomic technology. *IBM RedBook*, pages p. 5 – 9, 2005.

[Markl *et al.*(2003)]V. Markl, G. M. Lohman, and V. Raman. Leo : An autonomic optimizer for db2. *IBM Systems Journal*, Vol. 42, No. 1, 2003.

[Mateen *et al.*(2008)] A. Mateen, B. Raza, and T. Hussain. Autonomic computing in sql server. In *7th IEEE/ACIS International Conference on Computer and Information Science*, 2008.

[Nguyen *et al.*(2005)] T. M. Nguyen, J. Schiefer, and A. Min Tjoa. Sense & response service architecture (saresa). In *DOLAP'05*, 2005.

[Nicolicin-Georgescu *et al.*(2009)] Vlad Nicolicin-Georgescu, Vincent Benatier, Remi Lehn, and Henri Briand. An ontology-based autonomic system for improving data warehouse performances. In *Knowledge-Based and Intelligent Information and Engineering Systems, 13th International Conference, KES2009*, 2009.

[Saharia and Babad (2000)]A. N. Saharia and Y.M. Babad. Enhancing data warehouse performance through query caching. *The DATA BASE Advances in Informatics Systems*, Vol 31, No.3, 2000.

[Stojanovic *et al.*(2004)] L. Stojanovic, J. Schneider, A. Maedche, S. Libischer, R. Studer, Th. Lumpp, A. Abecker, G. Breiter, and J. Dinger. The role of ontologies in autonomic computing systems. *IBM Systems Journal*, Vol. 43, No. 3:p. 598–616, 2004.

RESUBMISSION: Management of Distributed Knowledge Sources for Complex Application Domains

Meike Reichle Kerstin Bach Alexander Reichle-Schmehl Klaus-Dieter Althoff

University of Hildesheim, Dep. of Computer Science
Intelligent Information Systems Lab
D-31141, Hildesheim, Germany
{lastname}@iis.uni-hildesheim.de

In order to realise a truly distributed knowledge-based system not only the knowledge processing step has to be carried out in a distributed way, but also the knowledge acquisition step. This paper's focus[1] lies on the distributed knowledge sources of the SEASALT architecture [Reichle *et al.*, 2009a] and their management and (optimised) querying using a Coordination Agent [Bach *et al.*, 2008]. Within SEASALT knowledge modularisation is realised in the *Knowledge Line* that is based on the principle of product lines as it is known from software engineering. We apply this to the knowledge in knowledge-based systems, thus splitting rather complex knowledge in smaller, reusable units (knowledge sources). Moreover, the knowledge sources contain different kinds of information as well as there can also be multiple knowledge sources for the same purpose. Therefore each source has to be described in order to be integrated in a retrieval process which uses a various number of knowledge sources.

A so called Knowledge Map organises all available knowledge sources that can be accessed by a Coordination Agent that creates individual requests and combines information. The term Knowledge Map originates in Davenport's and Prusak's work on Working Knowledge [Davenport and Prusak, 2000] in which they describe a Knowledge Map from the organisational point of view mapping human experts in a large organisation or company. We transfer this concept to an intelligent agent framework that coordinates different knowledge sources.

The Coordination Agent navigates through the Map and subsequently queries the individual knowledge sources and thus creating an individual path through the map[Reichle-Schmehl, 2008]. There are dependencies between knowledge sources, a dependency exists if one source's output serves as another's input and thus enforces a subsequent query. Since the dependencies between knowledge sources can take any form, the Knowledge Map is implemented as a graph where each knowledge source is represented by a node and directed edges denote the dependencies. Retrieval paths are computed based on the information a user gives in an individual query and the properties of the knowledge sources. Our current implementation provides an a-priori computation of the retrieval path using a modified Dijkstra algorithm to determine an optimal route over the graph.

Considering knowledge sources, different characteristics, and aspects on which to assess knowledge source properties come to mind. The possible properties can refer to content (e.g. quality or topicality) as well as meta-information (e.g. answer speed or access limits). In detail we have identified the following knowledge source (meta and content) properties.

- Meta properties: Access Limits, Answer Speed, Economic Cost, Syntax, Format, Structure, Cardinality, Trust or Provenance
- Content properties: Content, Expiry, Up-to-dateness, Coverage, Completeness

While these properties can be easily described and modeled, there are also more complex knowledge source properties. One of these more complex properties is quality: The quality of a knowledge source comprises many different aspects and we thus propose to also allow for compound properties to also permit the description of complex properties. Compound properties are the (weighted) sum of any number of the above presented simple topics.

Not all of the properties presented above are fully unrelated. The properties syntax, format, structure and cardinality for instance are partially related which allows for some basic sanity checks of their assigned values; also some of the properties such as answer speed, syntax or structure can be automatically assessed. Apart from these possibilities for automation the knowledge source properties currently have to be assessed and maintained manually by a Knowledge Engineer who assigns values to the properties and keeps them up to date.

References

[Bach *et al.*, 2008] K. Bach, M. Reichle, A. Reichle-Schmehl, and K.-D. Althoff. Implementing a Coordination Agent for Modularised Case Bases. In M. Petridis, editor, *Proc. UKCBR-08*, December 2008.

[Davenport and Prusak, 2000] T. H. Davenport and L. Prusak. *Working Knowledge: How Organizations Manage What they Know*. Harvard Business School Press, 2000.

[Reichle *et al.*, 2009a] M. Reichle, K. Bach, and K.-D. Althoff. The SEASALT Architecture and its Realization within the docQuery Project. In B. Mertsching, editor, *Proc. KI-2009*, LNCS, pages 556 – 563. Springer, 2009.

[Reichle *et al.*, 2009b] M. Reichle, K. Bach, A. Reichle-Schmehl, and K.-D. Althoff. Management of Distributed Knowledge Sources for Complex Application Domains. In K. Hinkelmann and H. Wache, editors, *Proc. WM2009*, LNI, pages 128–138, March 2009.

[Reichle-Schmehl, 2008] A. Reichle-Schmehl. Entwurf und Implementierung eines Softwareagenten zur Koordination des dynamischen Retrievals auf verteilten, heterogenen Fallbasen. BSc Thesis, September 2008.

[1] This is a one-page abstract for the full paper and references see [Reichle *et al.*, 2009b]

RESUBMISSION: An Extensible Semantic Wiki Architecture

Jochen Reutelshoefer, Fabian Haupt, Florian Lemmerich, Joachim Baumeister

Institute for Computer Science, University of Würzburg, Germany
email: {reutelshoefer, fhaupt, lemmerich, baumeister}@informatik.uni-wuerzburg.de

Abstract. Wikis are prominent for successfully supporting the quick and simple creation, sharing and management of content on the web. Semantic wikis improve this by semantically enriched content. Currently, notable advances in different fields of semantic technology like (paraconsistent) reasoning, expressive knowledge (e.g., rules), and ontology learning can be observed. By making use of these technologies, semantic wikis should not only allow for the agile change of its content but also the fast and easy integration of emerging semantic technologies into the system. Following this idea, the paper introduces an extensible semantic wiki architecture.

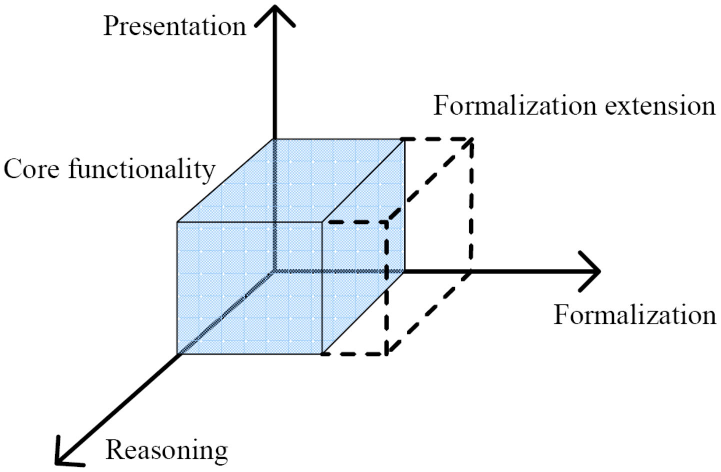


Fig. 1. The semantic wiki extension space: To support flexible knowledge engineering semantic wikis should be extensible along all the three dimension Formalization, Reasoning and Presentation

Resubmission: The original full paper has been published in: Proc. of SemWiki 2009 - The Semantic Wiki Web, Workshop co-located with the 6th European Semantic Web Conference, Crete, Greece (2009).

Task Patterns in Collaborative Semantic Task Management as Means of Corporate Experience Preservation

Uwe V. Riss, Benedikt Schmidt, Todor Stoitsev

{uwe.riss, benedikt.schmidt, todor.stoitsev}@sap.com
SAP Research, SAP AG
Dietmar-Hopp-Allee 16, 69190 Walldorf, Germany

Abstract

Experience Management in knowledge work has to take the execution of rather individual tasks into account. Therefore it is particularly important to provide a scheme that represents and allows to grasp the rather individual but reoccurring aspects in knowledge work. Task pattern approaches address this demand. In this paper we present two task pattern approaches: Personal Task Patterns focusing on individual task execution and resource handling and Collaborative Task Patterns for the support of process-related aspects. We show how both approaches can be combined and supplement each other.

1 Introduction

Experience sharing in knowledge work has been identified as a general challenge for knowledge management. This is mainly caused by the fact that knowledge work is largely self-determined, ad hoc, and often hidden, e.g., in email traffic [1]. Nevertheless, the demand for experience transfer in knowledge work is immense [2]. One reason is that the general fluctuation of employees in organizations requires the preservation and redistribution of execution knowledge to mitigate the risk of loss of experience since generally standard training for such work cannot be provided [3]. On the one hand, people who are new in an organization need to familiarize themselves with established proceedings. On the other hand, even domain experts often must solve unforeseen exception-to-the-rule problems. Even if best practices for extensive tasks such as projects are compiled after completion, such ex-post descriptions of tasks lack details and context reference that is required to fully understand the proceeding. Therefore methods that are more closely associated to actual task execution are required.

The most apparent way of transferring task experience seems to be the copying of task information as made available by some task management systems [12]. Instead of this direct task-to-task transfer we have suggested the use of task patterns to mediate the experience transfer [4]. Such task patterns originate from aggregation of individual task cases and provide information in the form of consolidated templates to users. One of the advantages of this approach is the clear separation of personal knowledge, which is contained in individual task data, and public task experience, which is represented in the task pattern. The separation helps motivate users to participate in the process of experience sharing by contributing to task patterns since their privacy remains respected [16].

Regarding the structure of task patterns, we have found that they can be either more information-oriented or more process-oriented. This distinction has led to the development of two kinds of task patterns which we call Personal Task Patterns (PTP) with a specific focus on the information aspect and individual execution [17; 21] and Collaborative Task Patterns (CTP) to support process-related aspects [22]. Although both approaches are based on the same principles and problem analysis, as described in [4; 15], they have been developed independently. In the current paper we want to describe how both types of task patterns complement each other and jointly allow for an extensive organizational experience capturing and reuse.

Both CTP and PTP apply a similar interaction schema. Users get support in executing their tasks, either with respect to collaborative aspects or with respect to information aspects, and in return they contribute to the enhancement of the used task patterns. This is based on the principle that task patterns provide guidance but always allow the users to deviate from the given suggestions. Moreover, users are encouraged to make their deviations public if this appears appropriate to them. The reuse and the incorporation of deviations in the task pattern increase its maturity. The enhanced task patterns are made accessible to other users without additional effort via a task pattern repository. The interplay of task pattern retrieval, use and enhancement realizes a task pattern lifecycle [6].

So far the two types of task patterns are embedded in different task management systems called Personal Task Management (PTM) and Collaborative Task Management (CTM), respectively. In the following we want to outline the central features of PTM and CTM and how they are supposed to work together. Some of the described features are already available, as can be seen from other publications [22; 21; 19]. Other features are planned to be realized in the project MATURE[1]. Here we give an outlook to such a system and describe what the user interaction will look like.

The paper is structured as follows. In Section 2 we give a short overview of the Personal Task Management and its implementation on the NEPOMUK Social Semantic Desktop[2] and introduce the PTP conception. In Section 3 we present the Collaborative Task Management and describe how it supports process planning in a knowledge work setting. This includes the presentation of CTP. In Section 4 we discuss the complementarity of Personal and Collaborative

[1] For further information about the project MATURE see: http://mature-ip.eu/en/start

[2] For further information about the project NEPOMUK see: http://nepomuk.semanticdesktop.org

Task Management that is reflected in a complementarity of the task patterns, respectively. In particular, we show how experience reuse can be realized based on the introduced concepts. In Section 5 we refer to related work that is relevant for the present approach before we finally discuss the results in Section 6.

2 Personal Task Management on the NEPOMUK Social Semantic Desktop

Task execution involves the complex interplay between information and work activities [7]. Although knowledge work is rather contingent it also includes reoccurring tasks and task schemes. In order to grasp these reoccurring features we have developed an infrastructure that supports their identification and management, the Personal Task Management implemented on the Semantic Task Management Framework (STMF) as the fundamental infrastructure [6]. This infrastructure is part of the NEPOMUK Social Semantic Desktop [8]. The integration of the PTM in a semantic framework allows the seamless semantic integration of information objects and task representations. This is a central precondition for a task pattern approach that aims at a consequent reuse of work-related information artifacts. The STMF serves as the foundation of semantic task management that offers task-related web services over the entire desktop.

As the central task management application that uses the STMF task services we have developed the KASIMIR sidebar for personal task handling [9]. It shows all tasks that are known to the STMF from whatever desktop applications they originate. In this way it provides a consolidated overview of all personal tasks. KASIMIR does not only give users an overview of their current tasks but allows them to assign basic properties, involved persons, used information objects, links, and subtasks to task. These task resources are information sources which the user regularly needs during task execution. They are represented in NEPOMUK semantic network and the semantic information can also be used by other desktop applications. A central goal of the task pattern approach has been to improve task execution by offering relevant information via Personal Task Patterns (PTP) and integrate this support directly in KASIMIR although in principle these task patterns are also accessible from other application.

PTP are structures that aim at the registration of all information artifacts used in a task. The information contained in PTP results from the executions of similar tasks which have been assigned to a pattern. Users can ignore suggested resources or add additional ones. In this respect the task pattern approach significantly differs from traditional workflow. Based on their experience and during the task execution users can supplement or modify the used PTP. Afterwards PTP can be exchanged between users or made public although the latter step is not mandatory. All deviations from the PTP are tracked by the STMF (as part of the personal desktop) in order to support users in updating PTP. This proceeding finally leverages a task pattern lifecycle [6].

2.1 Personal Task Pattern Details

The PTP recommends information assets by means of Abstraction Services. These are recommender services that suggest subtasks, information objects or persons to involve in the task. They describe the purpose of the recommended information object in the respective task context and provide descriptions of objects on different abstraction levels. In the current implementation they provide lists of objects, examples or conceptual description.

While the general descriptions and templates are mainly independent from the particular tasks, information objects and involved persons can depend on the particular context. The support of different abstraction levels is important since the offered information often require different degrees of abstraction: for a travel to Canada one needs to fill out specific forms (explicit recommendation) whereas the organization of the travel is often governed by more general aspects such as the bookmark of the internal travel request system or contact persons in case of problems (conceptual description).

PTP offer 4 different types of information: Exemplary Tasks, Abstraction Services, Decisions with Alternatives, as well as Problems and Solutions. In the following we point at some details while further explanations can be found in [21].

Abstraction Services:

The Abstraction Services provide information that helps users identify basic activities or resources to be used in their task. There are three basic types of Abstraction Services. (1) Information Abstraction Services and (2) Person Abstraction Services which guide activities on information objects and recommend possible collaborators required in the task context. (3) Subtask Abstraction Services suggest suitable subtasks that can be executed independently as proper tasks. As such they can make use of other task patterns. Thus Abstraction Services allow a guided decomposition of a task.

Decisions as filters:

Similar tasks often include similar decisions and action alternatives. The corresponding action alternatives consist of a set of Abstraction Services while the alternatives are realized by a filter function for the available Abstraction Services. The rationale of this filter functionality is that the decision for one of the alternatives redundantizes specific task options represented by the Abstraction Services. For example, in the business travel case a decision in a task pattern might offer the alternatives to travel by *plane* or *train*. Once the task performer has decided for *plane* the Abstraction Services related to *train* such as the bookmark to train connections are no longer offered to the task performer.

Problems and Solutions:

PTP also include certain problem statements [10] which occurred in tasks that used this pattern. Knowledge work is often characterized by unforeseen situations which require particular handling. Aid in these cases is especially needed if the task performer lacks experience. Therefore PTP provide a list of known problems specific for the type of task. Each problem can be associated to different solutions. Problems and solutions are given as short textual descriptions. The Solution additionally includes a description of the necessary activities. These activities are supported by suitable Abstraction Services. This allows the task performer to build up problem awareness and supports their solution.

2.2 From Tasks via Email to Collaboration

Task management in knowledge work requires the flexible collaboration with other users. This can even mean the transfer of metadata among collaboration partners to foster the semantic integration of the users' individual semantic networks and mutual understanding [19]. In this respect it is important that we give users the opportunity to (1) easily handle task and metadata that arrive via email and (2) create emails in which metadata can be easily included.

Within KASIMIR the delegation process is simplified by offering a *delegate a task* as individual task functionality. This helps users prepopulate an email with content and metadata that is available from the task. Users can directly select people to which they want to send the email from the list of persons involved in the task. With one click from the detailed task information they can open a selection screen that allows her to select those task resources that should be sent to the collaborators and an email form including copies of the resources and the respective metadata is opened.

By delegating or sharing tasks via email, the semantic annotations (e.g., tags, comments, bookmarks, etc) inherent to the task can be easily transferred to other project members. Since users often consider semantic annotations a tedious work, the sharing of semantic metadata via email can save a considerable amount of time. Incoming emails are checked for their relevance to the STMF and in case they originate from a STMF task context they are processed by the task framework and the metadata are incorporated in the NEPOMUK network. Users usually spend a significant amount of their workday filing and archiving incoming emails. However, decisions in which folder the information objects are to be stored, which name is appropriate and so on are fundamentally difficult regardless of the item being filed. Filing takes time and the folders that are created today may prove to be ineffective or even an impediment to the access of information in the future [18].

3 CTM Support for Collaboration

The Collaborative Task Manager (CTM) provides a different framework and supports other activities. It allows users to model, exchange and reuse lightweight, user-defined collaborative task structures. Thus, CTM concentrates on a structure to represent agile, human-centric business processes which emerge from tasks with attached subtasks, which again can contain subtasks and so on. Additionally information objects can be attached to these process structures for direct reuse. To support the handling of these processes the CTM system provides services to manually refine tasks in form of Collaborative Task Patterns (CTP) and realizes a process flow representation of tasks [14].

The CTM also provides general task management functionalities, extended by opportunities for process composition and collaboration via email. Support for process tailoring is provided by a close integration of the process definition in the actual user working environment. Emergent processes are grasped behind the scenes and generally made available for overview of evolving collaborative tasks and for reuse in recurring cases. The capturing and repeated execution of user activities in a software system is known as *"programming by example"* [24]. The CTM enables collaborative programming by example of agile process models, where users shape the emergent process in their area of expertise. In [22] we have described a framework for this lightweight composition of ad hoc business processes. It enables users to create hierarchical task lists by breaking down tasks into subtasks. Tasks are delegated via email. The recipients can further break down the received tasks and delegate the resulting subtasks to other end users. Changes of individual tasks in the user's personal task lists are tracked through web services on a central server instance where task data is replicated in a tracking repository. This means that the CTM is mainly a collaborative tool while privacy aspects play a minor role. It is the most natural procedure since the CTM is mainly concerned with collaborative processes. The tracking of the email exchange is made accessible as Task Delegation Graphs (TDG). TDGs represent agile process models that are captured as actual process execution examples and contain all task data including artifacts (attachments) and information on persons' involved in the process. TDGs enable informed participation of end users in process composition by providing a workflow-like overview of evolving collaborative tasks beyond the capabilities of common email and task lists.

The introduced framework enables the lifecycle management of agile process models through extraction, adaptation, and reuse of Collaborative Task Patterns (CTP) [6]. The CTP are designed as reusable task structure rather than information nodes, comprising tasks and their subtask hierarchies. They also store the complete context information of the contained tasks such as description, used resources, involved persons etc. CTP can be enacted to create a new process instance and execute it along the provided example activity flow. This flow can be altered by changing suggested task delegations or reusing referenced CTP hierarchies. CTP adaptation and reuse can result in evolution and complementation of captured process examples. This evolution is traced through task instance-based ancestor-descendant relationships [22]. In this way CTP enable users to establish best practices and to trace best-practice deviations in application cases.

3.1 Task Pattern in CTM

CTM realizes patterns on two granularity levels. On the one hand the TDG with their process-like overview, on the other hand the sharing of personal task lists. These patterns rely on a process of iterative adaptation and reuse which can result in refinement of captured process examples. CTM enables tracing of evolving patterns through task instance-based ancestor/descendant relationships [11]. These are iteratively generated between the tasks in the originating hierarchy and the corresponding tasks in the resulting hierarchy when a task hierarchy is reused.

These task lists shared as pattern do not include an abstraction mechanism as we find it in the STMF. They rely on example-based transfer of task execution knowledge. Thus, the strength of patterns in the CTM lies in the representation of personal task hierarchies of different users and their relationships resulting from former delegation flows.

3.2 CTM Workflow Modeling and STMF Integration

In CTM, the captured pattern evolution is used to detect regularly occurring process fragments. The CTM enables transformation of user-defined TDG to formal workflows. This bridges ad hoc and formal process representations. The resulting graphs show non-formalized, ad hoc user behavior which is not constrained by formal business rules. The formalizing transformation process requires a domain expert who is able to use the given data as basis for a better understanding and a modeling based on empirical data. The degree, to which the generated formal workflow models require correction or complementation, depends on how the users work with ad hoc CTM tasks.

With respect to an integration of CTM and STMF we plan the following proceeding: The STMF already supports the delegation of subtasks via email. Therefore it is planned to use the existing user interface in KASIMIR to create CTM tasks. Thus, the CTM is just regarded as another desktop application making use of the STMF web services. When delegating a subtask, users select subtask metadata which are to be sent with the CTM tasks. These tasks describe independent but subordinate activities as part of the task owned by the sender. Therefore they appear as proper tasks on the recipient's side. The integration is facilitated by the fact that all NEPOMUK Semantic desktops share a basic Personal Information Model Ontology (PIMO) [13] to describe information objects. This is a precondition for the sharing of metadata within project teams.

The STMF also reads the tasks received in the CTM and transfers them to the NEPOMUK repository. The corresponding metadata are included so that the user does not only get the task data but also the semantic graph related to the email. This disburdens them from the task to additionally annotate the received attachments.

4 Complementarity of PTM and CTM

Summing up, the PTM has been designed as a system for personal task management on the desktop whereas the CTM has been designed as a collaborative task management system that focuses on the relations between tasks and the joint use of resources. The transition of tasks from the PTM to the CTM can be seen as the transition from the private to the public sphere. This is emphasized by the fact that tasks are *manually* transferred from the PTM to the CTM as a conscious act of publication. This allows the user to decide which part of the so far private task-related information is transferred to the CTM.

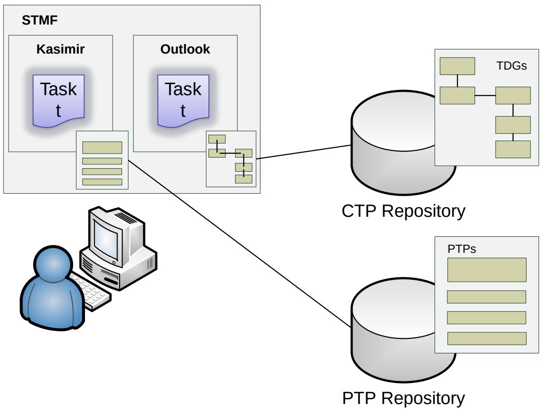


Figure 1: User perspective on how KASIMIR and CTM work together

In Figure 1 we show the user perspective of the interplay of PTP, which are accessible via the KASIMIR task management application, and CTP, which are accessible via Microsoft Outlook. While users transfer tasks between both systems they can make use of the respective task patterns in each system. This means that in Outlook, where they deal with the delegation of tasks, they get support with respect to underlying process. In KASIMIR such process aspects are not necessary and will not be offered. Both KASIMIR and the Outlook Add-In of CTM use the task services offered by the STMF to exchange task data.

Although PTM and CTM address the same problem [4] the different design orientations influences the way how the respective task patterns are defined. While the CTP addresses the collaborative, i.e., process related, aspects, the PTP concentrates on the personal task handling including specific use of resources and chronological order of their usage, i.e., the focus on the personal reuse of experience gained in particular executions of tasks. The guidance provided by the Abstraction Services is only one example how this personal guidance is realized.

Email is an established and safe way of information exchange in enterprises. It is generally available and mainly platform-independent. Email also supports the transfer of metadata among collaboration partners. In this way it fosters the semantic integration of the individual semantic networks of different users. The importance of the possibility to (1) easily handle tasks and metadata that arrive via email, and (2) to create emails in which metadata can be easily included has led to the decision to choose email as delegation channel. The coupling between PTP and CTP is realized via individual tasks. When tasks are transferred between PTM and CTM the relationship between them becomes part of the task data. When a PTM task becomes a CTM task, the PTM task stores the identifier of the CTM task so that a later identification is possible, e.g., when the delegated task is completed.

PTP and CTP together cover the entire spectrum of work experience that can be reused. They provide the means of extensive experience handling including the following aspects:

- *Collaboration:* The CTP describes all aspect of collaboration and division of labour. This does not only hold for individual delegation steps but for entire process chains with repetitive delegation.
- *Resource Handling:* The PTP describes which resources have been used in specific kinds of tasks and their role using Abstraction Services.
- *Problem Solving:* The PTP describes individual problems that occurred during the execution of specific task types. This information helps other users to find solutions when they face similar problems.
- *Decisions:* The PTP supports users in making typical decision situations in tasks transparent and offering the known alternatives. This functionality can show existing alternatives and helps users to recognize branching points in the execution.

The complementarity of task patterns also solves a specific problem of PTP. Since they possess a mainly private character they are not generally accessible in organizational repositories. This raises the questions how users get access to them. Let us assume that a user receives a CTM task that she accepts and transfers into her PTM. So far no PTP is used. However, the PTM can now send a request to the CTM in order to find task patterns that were previously used for this kind of task. So far the users do not get access to suitable PTP which might reside on the desktops of other users. However, if they possess a personal relation to some of these users, e.g., via a network of trusted relationships, they are able to ask the specific user for access. Of course there might be more possibilities how such a relationship can be established but this seems to be a straightforward way.

The advantage for users of the PTM is that the integrated system allows them to delegate tasks via email and also to exchange metadata this way. Such collaborative requirement can be considered as mandatory for an organizational task management system. On the other hand, the users of the CTM obtain the possibility to integrate task related information in their personal semantic network as it is provided by the NEPOMUK social semantic desktop. The semantic desktop enables them to make use of task information in their personal information management [20] and helps them to keep it accessible instead of loosing track of it in some task management silo.

The advantage of this approach is that it is not only possible to transfer rather abstract and high level information but also very detailed and rich information. Moreover, the requesting user finds a contact person with the provided information while at the same time this contact person is disburdened by the richness of the data which often avoids asking additional questions. Thus, both parties profit from the transaction.

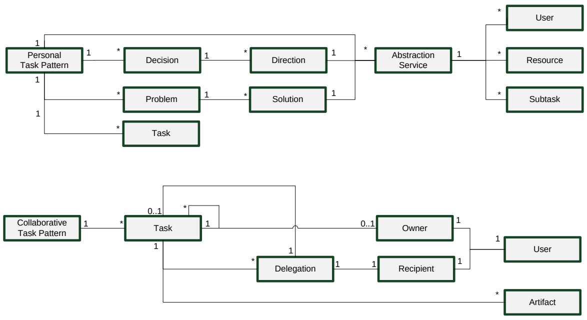


Figure 2: Main task pattern elements for PTP and CTP

In Figure 2 we have described the main elements of PTP and CTP. For PTP the reference to all users, resources, and subtasks is mediated by Abstraction Services in order to group objects with the same role and allow context dependent access. In this way the access to these objects becomes more comprehensive to the user [21]. Moreover, PTP use the Abstraction Services to support higher level entities such as Decisions and Problems. Besides, also the direct use of Abstraction Services is possible. In contrast, CTP take tasks and their procedural relationships as the central aspects. Related to this process is the fact that delegation and ownership are explicitly considered. On the other hand, artifacts are only considered as simple resources that are general usable. Details of the CTP model can be found in [22; 23].

4.1 Task Pattern Evolution

The decisive point for experience management is the continuous improvement of task patterns. In both CTM and PTP, the experience management is based on the recognition of users' deviations from best-practice suggestions. Both task management systems provide means to trace such deviations through task instance-based relationships [21; 22], even if PTP and CTP focus on the individual and collaborative aspects, respectively.

When a CTP is exported from an executed process and saved to a remote task patterns repository, all resulting tasks receive ancestor references to the corresponding original tasks in the tracking repository. If a remote CTP is applied, the resulting tracked tasks receive ancestor references to the corresponding tasks of the remote CTP.

In the PTM the update of PTP works in a similar way. KASIMIR provides functionalities to maintain PTP and related objects like Abstraction Services, Decisions with Directions and Problems with Solutions. It allows the automatic transformation of task instances to PTP. Therefore a detailed overview of the connected tasks and their use contexts is provided to the user. Additionally classes of attached objects optionally in a task journal view which shows their temporal relation as presented in [10] is provided. Such additional task information gives an overview of the different tasks belonging to the task classes described by a PTP.

The general interplay between PTP and CTP can be described as follows. The CTP provides experience in terms of process knowledge, e.g., hierarchies of tasks, people to which tasks can be delegated and so on. They are generally public and provide an overview of the global processes that take place. Experience regarding the execution of an individual task, however, are rather stored in PTP. This concerns resources that might be used, bookmarks, or subtasks that are executed by the users themselves. The nature of the PTP is mainly private and they provide detailed information that might be partially requested from individual users.

5 Related Work in Task-Oriented Experience Sharing

The support of task execution in knowledge work has been in the focus of different research projects. Many of these approaches deal with the question how to share experience by means of information captured in task-centric information systems. We find two different types of experience sharing. The first approach exploits task objects with subtasks and attached information objects as templates; the second approach uses abstractions of task information to realize process-like representations of task execution.

The Task Navigator [5] combines task-oriented proactive information delivery and tool support for task management. Proactive information delivery means recommendations of information objects based on former task execution. Additionally users can make use of tasks as templates or use process-like description of tasks (process-types). Proposals and the reuse of tasks and process types are embedded as recommendation lists into tree-like task structures.

An approach to support task execution by non-prescriptive guidelines has been proposed in the context of the Unified Activity Management project (UAM) [25]. Activity Patterns have been proposed as dynamic structure to disseminate task execution knowledge in terms of activities. Aspects of this concept have been realized as activity templates in Lotus Notes Activities [26]. Activity templates can be created by domain experts or be created based on earlier task execution. They describe activities necessary to execute a task. The template provides placeholders which stand for objects used in the task context. Placeholders lack the contextualization of information objects on different abstraction levels. Additional knowledge collected by the community requires additional maintenance effort to be included in the activity template. Although, the direct reuse of such knowledge to enhance an activity pattern has been proposed in [27] it has not been realized beyond the presented paper-based user study.

The given overview shows the difficulty to find a useful abstraction level for task execution support. Templates based on individual tasks are often too case specific, while process-like descriptions are too formal. Such templates give users support in form of listed subtask and information object recommendations for task cases and complicate identification of purpose and context. Process models are represented by complex flow-charts which fail to adequately represent the flexible nature of task execution and often overstrain the individual user. Different support techniques combined in one system complicate the retrieval and reuse of task execution knowledge. The presented approach avoids such differences by relying on one single concept: Task Patterns with Abstraction Services which can model different abstraction degrees, as Abstraction Services are capable to include conceptual information, information objects and information retrieval support. The Activity Pattern approach of the UAM project provides an alternative of task knowledge reuse and abstraction without relying on process descriptions, but lacks a concept to distribute this knowledge to the user and a realization as software.

With respect to the process oriented solutions we find further approaches. One comprehensive approach, addressing the gap between completely ad hoc processes, which are in the focus of Computer Supported Cooperative Work (CSCW), and rigid, predefined business processes, which are well supported through conventional workflow solutions, is provided by Bernstein [28]. This approach provides “contextual basis for situated improvisation” by enabling delivery of “process models, process fragments, and past cases” for tasks and providing shared, distributed-accessible, hierarchical to-do lists, where different process participants can access and enrich task resources and information. An extended state of the art study in the area of flexible workflows and task management and a further approach for integrating ad hoc and routine work is presented by Jorgensen [29]. He reveals major issues concerning business process flexibility and how it can be facilitated through interactive processes models.

Approaches focusing on completely ad hoc processes are also known. A case-based approach for enabling business process flexibility, where ”the knowledge worker in charge of a particular case actively decides on how the goal of that case is reached” is provided by van der Aalst et al. [30]. A further solution of supporting completely ad hoc processes is presented by Holz et al. [5]. It provides document-based and task-based proactive information delivery, which enables evaluation of similar cases and instance-based task reuse. Thereby it is suggested that frequently recurring tasks, relevant for an enterprise, are modeled more formally using process types if the enterprise is willing to make an investment into process modeling. Advanced techniques for building personal knowledge spaces and wiki-based collaborative document spaces are also integrated in the latter solution.

The major difference of the frameworks presented in this paper to the above mentioned approaches is that they focus on the unobtrusive support for ad hoc business processes. They enable users to act as close as possible to their usual work practices without confronting them with new working environments or upfront process definition tools. CTM unfolds emergent process structures meanwhile PTM supports the information oriented task execution. The motivation behind this approach is that enterprise processes are generally executed by multiple actors, who have different level of technical skills and different attitude towards maintaining process data. At the same time analysis, reuse and adaptation of knowledge-intensive processes is often desired in a way similar to conventional workflows. The framework therefore enables end users without advanced technical expertise or process understanding to manage tasks in personal task lists, which are integrated in a common software working environment. As such the framework uses email, which plays a central role for the exchange of tasks and task-related information in organizations [22]. Behind the scenes, personal task hierarchies of multiple process participants are reconciled to overall enterprise processes in central repositories, where context information and resources are provided on-demand to advanced users and process analysts. Thereby no formal process modeling, explicit definition of rules or user roles is required.

6 Conclusions

We have presented an approach that integrates two task pattern methods for experience management in knowledge work. While the first, CTM, focuses on process aspects, the second, PTM, concentrates on the reuse of individual work experience. Whereas the CTM addresses public interaction patterns, the PTM tries to grasp rather private and rich task execution data. Both have in common that they are based on task patterns as means to transfer experience, called CTP and PTP, respectively. The private character of PTP becomes manifest in the fact that they only reside on the users’ personal semantic desktop. This privacy is important for the acceptance by users. Nevertheless PTP can be exchanged on a trusted person-to-person relationship. In contrast, CTP are designed as public data so that they are generally available. Thus, we have proposed to use networks of trusted relationships to support the distribution of PTP mediated by CTP. This allows users find CTP and use the relations between CTP and PTP to find and to request access to the related PTP via trusted connections. This seems to be a feasible way to spread the personal information contained in PTP without violating the users’ privacy.

The fundamental advantage of this approach is that it deals with very rich task information which is furthermore closely integrated in the task execution process. Both CTM and PTM aim at the provision of task related information directly incorporated in the specific task management environment. In this way users cannot only *read* best practice but they can directly apply it in their own tasks.

In this way we avoid that work experience must first be transformed into an abstracted form which afterwards must be concretized in task execution again. During this process of abstraction and concretion a considerable part of the information gets lost. Often exactly these details contain the really valuable information for experience management. Another problem is that these details often get lost in the course of the task execution or that they are not considered as relevant by those who later write best practice reports. In any case it is easier to record the respective information in the course of task execution than ex-post.

An evaluation of the integrated system is planned on the basis of a comparison to PTM as well as CTM, respectively. The focus of such evaluation is to investigate whether the users of the respective systems experience the additional functionality as enrichment of their current task management systems.

The integration is planned as part of the project MATURE that aims at the realization of knowledge maturing processes in organizations and networks of organizations. The task pattern idea reflects the core idea of MATURE with respect to work experience from an organizational perspective since considerable knowledge is developed by individual users and must be preserved and further developed to gain its full potential in an organization. Task patterns support this process by collecting individual experience and make it available to many others. They foster the consolidation of experience since they bring together the experience of various users who are motivated to contribute their part to the collective memory in an unobtrusive way.

References

[1] R. McDermott. Working in public - learning in action: designing collaborative knowledge work teams. Advances in Interdisciplinary Studies of Work Teams, 2, 1995.

[2] J. Conklin. Designing Organizational Memory: Preserving Intellectual Assets in a Knowledge Economy. Group Decision Support Systems, 1, 1996.

[3] E.W. Stein. Organization memory: Review of concepts and recommendations for management. International Journal of Information Management, 15(1), 17-32, 1995.

[4] U. V. Riss, A. Rickayzen, H. Maus, W.M.P. van der Aalst. Challenges for Business Process and Task Management. Journal of Universal Knowledge Management, vol. 0(2), 77-100, 2005.

[5] H. Holz, O. Rostanin, A. Dengel, T. Suzuki, K. Maeda, T. Kanasaki: Task-based process know-how reuse and proactive information delivery in TaskNavigator. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, 522-531, 2006.

[6] E. Ong, O. Grebner, U.V. Riss. Pattern-based task management: pattern lifecycle and knowledge management. In 4 thConference of Professional Knowledge Management (WM 2007), Workshop Integrierte Wissensmanagement-Systeme (IKMS2007), Potsdam, Germany, 2007.

[7] U.V. Riss. Knowledge, Action, and Context: Impact on Knowledge Management. LNCS, vol. 3782, 598-604, 2005.

[8] T. Groza, S. Handschuh, K. Moeller, G. Grimnes, L. Sauermann, E. Minack, C. Mesnage, M. Jazayeri, G. Reif, R. Gudjonsdottir. The NEPOMUK Project-On the way to the Social Semantic Desktop. In 3rd International Conference on Semantic Technologies (I-Semantics 2007), Graz, Austria, 2007.

[9] O. Grebner, E. Ong, U.V. Riss. KASIMIR-Work process embedded task management leveraging the Semantic Desktop. pp. 1715-1726, 2008.

[10] U. V. Riss, O. Grebner, Y. Du. Task Journals as Means to Describe Temporal Task Aspects for Reuse in Task Patterns. In: ECKM 2008 Proceedings of the 9th European Conference on Knowledge Management. Southampton, UK, 721-730, 2008.

[11] T. Stoitsev, S. Scheidl, M. Spahn. A Framework for Light-Weight Composition and Management of Ad-Hoc Business Processes. In: Winckler, M., Johnson, H., Palanque, P. (eds.) TAMODIA 2007. LNCS, vol. 4849, 213-226, 2007.

[12] H. Holz, A. Dengel, T. Suzuki, K. Kanasaki. Task-based process know-how reuse and proactive information delivery in TaskNavigator. In Proceedings of the 15th ACM international conference on Information and knowledge management. 2006.

[13] L. Sauermann, L. v. Elst, A. Dengel. PIMO - a framework for representing personal information models. In: T. Pellegrini, S. Schaffert (eds.) Proceedings of I-MEDIA '07 and I-SEMANTICS '07 International Conferenceson New Media Technology and Semantic Systems as part of TRIPLE-I 2007, J.UCS, 270-277, 2007.

[14] T. Stoitsev, S. Scheidl, F. Flentge, M .Mühlhäuser. Enabling end-user driven business process composition through programming by example in a Collaborative Task management system. IEEE Symposium on Visual Languages and Human-Centric Computing, 157-165, 2008.

[15] H. Holz, H. Maus, A. Bernardi and O. Rostanin. From Lightweight, Proactive Information Delivery to Business Process-Oriented Knowledge Management, Journal of Universal Knowledge Management, vol. 0(2), 101-127, 2005

[16] U.V. Riss, U. Cress, J. Kimmerle, S.Martin. Knowledge transfer by sharing task templates: two approaches and their psychological requirements. Knowledge Management Research & Practice. 5(4), 287-296, 2007.

[17] Y. Du, U. V. Riss, E. Ong, P. Taylor, Liming Chen, D. Patterson, Hui Wang. Work Experience Reuse in Pattern Based Task Management. I-KNOW, Graz, forthcoming, 2009.

[18] W. P. Jones, H. Bruce, S. T. Dumais. Keeping found things found on the web. In: Proceedings of the 10th ACM International Conference on Information and Knowledge Management (CIKM'01). ACM, 119-126, 2001.

[19] U. V. Riss, M. Jurisch, Viktor Kaufman. Email in Semantic Task Management. CEC 2009, Vienna, forthcoming,. 2009.

[20] J. Teevan, W. Jones, B. B. Bederson. Personal Information Management. Communications of the ACM, 49(1), 40-43, 2006.

[21] B. Schmidt, U. V. Riss. Task Patterns as Means to Experience Sharing. ICWL, Aachen, forthcoming, 2009.

[22] T. Stoitsev, S. Scheidl, M. Spahn. A Framework for Light-Weight Composition and Management of Ad-Hoc Business Processes. LNCS, vol. 4849, 213-226., 2007.

[23] T. Stoitsev, S. Scheidl. A Method for Modeling Interactions on Task Representations in Business Task Management Systems. In: P. Forbrig and F. Patern (Eds.): HCSE/TAMODIA 2008, LNCS 5247, 84-97, 2008.

[24] H. Lieberman. Your wish is my command: Programming by example. Morgan Kaufmann, 2001.

[25] T.P. Moran, A. Cozzi, S.P. Farrell. Unified activity management: supporting people in e-business. Communications of the ACM, 48(12), 67-70, 2005.

[26] W. Geyer, M. J. Muller, M. Moore, E. Wilcox, L. T. Cheng, B. Brownholtz, C. Hill, D. R. Millen. ActivityExplorer: Activity-Centric Collaboration from Research to Product. IBM Systems Journal, 45(4):713-738, 2006.

[27] G. Convertino, T.P. Moran, B.A. Smith. Studying activity patterns in CSCW, 2007.

[28] A. Bernstein: How Can Cooperative Work Tools Support Dynamic Group Processes? Bridging the Specificity Frontier. In: CSCW 2000, ACM Press, New York, 279-288. 2000.

[29] H. d. Jorgensen: Interactive Process Models. Ph.D. Thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2004.

[30] W.M.P. v.d. Aalst, M. Weske, D. Grünbauer: Case Handling: A New Paradigm for Business Process Support. Data and Knowledge Engineering, 53(2), 129-162, 2005.

Content-sensitive User Interfaces for Annotated Web Pages

Florian Schmedding
University of Freiburg, Germany
schmeddi@informatik.uni-freiburg.de

This is a resubmission of work published at the AST Workshop 2009 [Schmedding, 2009].

1 Introduction

A broad range of different semantic technologies have been developed and standardized over the last years, but browsing on web sites rarely takes advantage of them although semantic descriptions can be supplied with RDF. We see one reason for it in the missing linkage between this data and the page content being presented to the user. Consequently, the underlying data cannot be addressed via an interaction with the visual representation. Despite having already found what one was looking for in the document, selecting the according data in a separate location is necessary for further automatic processing.

Recently, a new formalism—RDFa [Adida and Birbeck, 2008]—that seeks to close this gap has been standardized by the W3C. It defines some additional attributes for XHTML and a mapping from the attribute values and the document content into RDF triples. An important feature in our context is the usage of selected text parts for literal properties and hence the explicit linkage between the human- and the machine-readable representations. In contrast to the mapping, the handling of this linkage is left open in the reference specification.

2 Contribution

In this work, we will introduce a novel way of handling this linkage within the document object model and show its utility for several new use cases for RDFa documents. We propose an integration of the RDF statements into the document object model of the XHTML pages as a generic way to manage RDFa annotations. Covering resemblances between the use cases it makes their handling easier.

In particular, we focus on applications and user interfaces that relate the annotations on web pages to knowledge from other sources. This means they operate in two directions: extracted data from the page can be forwarded and response data can be integrated right into the displayed page in the browser, close to the respective annotated elements. Of course, this approach is not limited to web browsers but applies to HTML documents in general. For instance, it could be used also for HTML e-mails.

3 Extension of the Document Object Model

With our integration of RDFa and DOM we are targeting the visual representation of web documents. As the interaction with web pages regards the DOM representation, the extension of this model with functions to manage RDF statements seems a natural derivative. Our approach makes it possible to retrieve DOM elements by SPARQL queries; DOM elements in turn provide information about the contained statements and resources. Therefore we add properties to store subject, predicates, objects, datatype, and type information of RDFa to each XML element; for the statements we include a triple store. Similar to existing DOM-methods, we provide methods to retrieve XML elements that contain a given resource or statement. More specific conditions can be expressed by SPARQL queries. Instead of variable bindings, the respective method returns the XML elements where the resulting resources are defined. The necessary linkage information is contained directly in the data structures of the used triple store.

4 Applications

In our use cases, we use RDFa not only in one but in two directions: we let the web browser extract data *and* receive further data from other sources to provide advanced feedback for users. Knowing the linkage between text and meaning is especially necessary in the second case.

For example, *appointments* on a web page can be compared to entries in the user's personal calendar. To notify the user about overlaps, the passages describing conflicting dates are highlighted. Thus one only perceives feedback for content one is actually reading. Additional actions can be provided via context-menus on the respective elements. Similarly, *input suggestions* can be supplied, e. g. when entering a surname into an annotated form of an online telephone directory: the user's address book can be searched for matching forenames, expecting better results.

References

[Adida and Birbeck, 2008] Ben Adida and Mark Birbeck. RDFa Primer. `http://www.w3.org/TR/2008/NOTE-xhtml-rdfa-primer-20081014/`, 2008.

[Schmedding, 2009] Florian Schmedding. Content-sensitive User Interfaces for Annotated Web Pages. In *Proceedings of the 4th Int'l Workshop on Applications of Semantic Technologies (AST)*, Lübeck, Germany, 2009. *To appear*.

The Neighborhood Graph for Clinical Case Retrieval and Decision Support within Health-e-Child CaseReasoner

Alexey Tsymbal, Gabor Rendes, Martin Huber
Corporate Technology Division
Siemens AG, Erlangen, Germany
{alexey.tsymbal; gabor.rendes;
martin.huber}@siemens.com

Shaohua Kevin Zhou
Integrated Data Systems Department
Siemens Corporate Research
Princeton, NJ, USA
kzhou@scr.siemens.com

Abstract

In the context of the EU FP6 project Health-e-Child, a Grid-based healthcare platform for European paediatrics is being developed. The basic philosophy behind the design of CaseReasoner, a similarity search based decision support and knowledge discovery system we are developing for Health-e-Child, is to provide a clinician with a flexible and interactive tool to enable operations such as data filtering and similarity search over a Grid of clinical centres, and also to facilitate the exploration of the resulting sets of clinical records regardless of their geographical location. In order to visualize patient similarity, besides the more orthodox heatmaps and treemaps a novel technique based on neighborhood graphs is being developed, which is in the focus of the present paper. For similarity search on distributed biomedical data, besides the canonical distance functions novel techniques for learning discriminative distance functions are also made available to the clinician. The use of distance learning techniques in combination with the patient similarity visualization modules of CaseReasoner contributes to making it a powerful tool for clinical knowledge discovery and decision support in various classification contexts; it helps to combine the power of strong learners with the transparency of case retrieval and nearest neighbor classification.

1 Introduction

There is growing interest in the use of computer-based clinical decision support systems (DSSs) to reduce medical errors and to increase health care quality and efficiency [Berlin *et al.*, 2006]. Clinical DSSs vary greatly in design, functionality, and use. According to the reasoning method used in clinical DSS, one important subclass is that of Case-Based Reasoning (CBR) systems – systems which have reasoning by similarity as the central element of decision support [Berlin *et al.*, 2006; Nilsson and Sollenborn, 2004].

One reason for the slow acceptance of CBR systems in biomedical practice is the especial complexity of clinical data and the resulting difficulty in defining a meaningful distance function on them and adapting the final solution [Schmidt and Vorobieva, 2005]. Another commonly reported reason for the relatively slow progress of the field is the lack of transparency and explanation in clinical CBR. Often, similar patients are retrieved and their diagnoses are presented, without specifying why and to what extent the patients are chosen to be similar and why a certain decision is suggested. We believe that, one way to approach this problem is to better visualize the underlying inter-patient similarity, which is the central concept of any clinical CBR.

In known CBR systems the visualization is usually limited with the visualization of case solutions and not case similarity [Mullins and Smyth, 2001]. To solve the problems described above, we introduce a novel technique for visualizing patient similarity, based on neighborhood graphs, which can be helpful in clinical knowledge discovery and decision making. Besides, we consider two related techniques for learning discriminative distance functions, which when used in combination with the neighborhood graphs can make them a powerful and flexible tool for clinical decision making in different classification contexts.

In this paper we introduce a novel technique for visualizing patient similarity, based on neighborhood graphs; we also discuss the architecture of our implementation within the Health-e-Child DSS CaseReasoner and in particular the related techniques for learning discriminative distance function. The main advantage of the suggested technique is that the decision support becomes *transparent*. The nearest cases and the underlying similarity used for decision making can easily be visualized with the three types of neighborhood graphs. Moreover, after replacing the commonly used “black box” classification with distance function learning and case retrieval, the accuracy of classification usually remains same or even becomes better, and there appears a possibility to visualize the nearest cases of suggested class (say, malignant) and nearest cases of the other class (say, benign), in order for the user (clinician) to analyse and double-check the decision suggested.

The similarity search-based clinical knowledge discovery and decision support system CaseReasoner, besides neighborhood graphs also uses treemaps [Shneiderman, 1992] and heatmaps in order to better represent inter-patient similarity [Tsymbal *et al.*, 2007a]. In particular, the treemap and the heatmap in CaseReasoner represent a hierarchical clustering of patients obtained based on a certain distance function defined by a clinician e.g. via a set of data attributes of interest or a distance function previously learnt for a certain classification context.

The work in our study has been performed as part of the Health-e-Child (HeC) project. HeC is an EU-funded Framework Programme 6 (FP6) project, which was started in 2006, and aims at improving personalized healthcare in selected areas of paediatrics, particularly focusing on integrating medical data across disciplines, modalities, and vertical levels such as molecular, organ, individual and population. The project of 14 academic, industry, and clinical partners aims at developing an integrated healthcare platform for European paediatrics while focusing on some carefully selected representative diseases in three different categories; paediatric heart diseases, inflammatory diseases and brain tumours. The material presented in this paper contributes to the development of decision support facilities within the platform prototype which provide the clinicians with tools to easily retrieve and navigate patient information and help visualizing interesting patterns and dependencies that may lead, besides personalized decision making concerning appropriate treatment, to establishing new clinical hypotheses and ultimately discovering novel important knowledge.

The paper is organized as follows. In Section 2 the technique of patient similarity visualization based on neighborhood graphs is considered, our implementation of it is discussed and a few examples are given. Section 3 presents techniques for learning discriminative distance functions which can be used to learn a strong distance function in different contexts and which nicely complements the patient similarity visualisation techniques. Section 4 presents the overall architecture of the similarity-search based decision support and knowledge discovery system CaseReasoner, and in Section 5 a few related open issues are discussed with a focus on its evaluation. We conclude in Section 6 with a brief summary, open issues and further research topics.

2 Neighborhood Graphs

2.1 Introduction and Related Work

Neighborhood graphs provide an intuitive way of patient similarity visualization with a node-link entity-relationship representation. There can be distinguished three basic types of neighborhood graphs that can be used to visualize object proximity in DSSs; (1) relative neighborhood graph (RNG), (2) distance threshold graph, and (3) directed nearest neighbor graph. These graphs are studied and applied in different contexts; in particular, as data visualization tools the threshold and nearest neighbor graphs are often used for the analysis of gene expression data in bioinformatics [Zhang and Horvath, 2005; Scharl and Leisch, 2008]. Thus, Zhang and Horvath [2005] study so-called gene co-expression networks, which are represented with the threshold neighborhood graph. Scharl and Leisch [2008] suggest using the nearest neighborhood graph in order to visualize gene clusters.

In a *relative neighborhood graph*, two vertices corresponding to two cases A and B in a data set are connected with an edge, if there is no other case C which is closer to both A and B with respect to a certain distance function *d* [Toussaint, 1980]:

$$d(A,B) \le \min_{C \ne A,B} \max\{d(A,C), d(B,C)\} \qquad (1)$$

Originally, relative neighborhood graphs were defined for 2D data (planar sets) with the Euclidean distance metric, but later they were generalized and applied to multiple dimensions and other distance functions [Toussaint, 1980; Jaromczyk and Toussaint, 1992; Muhlenbach and Rakotomalala, 2002].

Besides the relative neighborhood graphs we focus on, there are known some other related node-link (graph-based) visualizations of instance proximity. These include the Minimum spanning tree (MST), the Gabriel graph, and the Delanay tessellation [Jaromczyk and Toussaint, 1992]. We believe that out of this family, the relative neighborhood graph is the best candidate to visualize patient proximity in a DSS. The MST has usually too few edges to spot groupings/patterns in the data, while the Gabriel graph and the Delanay tessellation are, vice versa, usually too overcrowded, which becomes a problem with already more than a hundred cases (patients).

A *threshold* graph is simply defined as a graph where two vertices are connected with an edge if the distance between the two corresponding cases is less than a certain threshold. In a *nearest neighbor graph*, each case is connected with one or a set of its nearest neighbors. This graph is usually directed as the relation of being a nearest neighbor is not necessarily symmetric. An important benefit of RNG comparing to the other two graphs is the fact that it is always connected with nodes having a reasonable small degree; it is often planar or close to planar.

In machine learning, neighborhood graphs find various applications, including data clustering, outlier removal, and even supervised discretization [Muhlenbach and Rakotomalala, 2002]. The *k*-nearest neighbor (*k*-nn) graph is often used as the base in various approximate nearest neighbor search techniques in high-dimensional spaces, in order to cope with the curse of dimensionality, see [Sebastian and Kimia, 2002; Paredes and Chavez, 2005] for two examples. Besides, neighborhood graphs may serve as a source of measures of complexity for such searching, in order to estimate the costs [Clarkson, 2006]. Another important related branch of research studies how to optimize the process of construction of the neighborhood graph. Thus, Paredes *et al.* [2006] optimize construction of the *k*-nearest neighbor graph in metric spaces and achieve empirically around $O(n^{1.27})$ complexity in low and medium dimensional spaces and $O(n^{1.9})$ in high dimensional ones. [Jaromczyk and Toussaint, 1992] review algorithms for reducing the complexity of constructing an RNG, which is $O(n^3)$ in general, in low-dimensional spaces.

Besides machine learning and similarity search optimization, in other domain areas other, more exotic applications may also be found. In [Marcotegui and Beucher, 2005], for example, the minimum spanning tree of a neighborhood graph was applied to contrast-based hierarchical image segmentation. In [Li and Hou, 2004] a directed relative neighborhood graph and directed minimum spanning tree are successfully applied to topology control, in order to create a power-efficient network topology in wireless multi-hop networks with limited mobility.

Fig. 1 below presents an example of a neighborhood graph constructed for the Leukemia public gene expression data set (available at www.upo.es/eps/aguilar/ datasets.html) within CaseReasoner. RNG for a set of 72 samples representing healthy (blue) and diseased (red) patients is shown. The underlying distance function is the intrinsic Random Forest distance. Leave-one-out accuracy for this problem is as high as 98%. Such a graph provides a powerful tool for knowledge discovery and decision making in the considered domain (e.g., by placing and displaying the gene expression sample of a new patient with unknown diagnosis in such a graph).

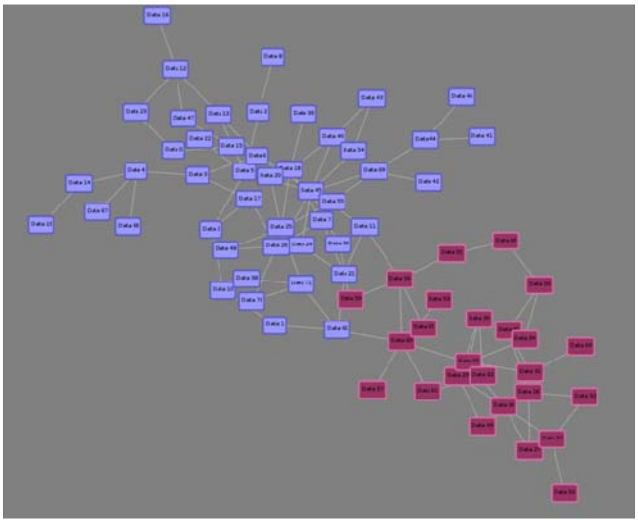


Figure 1 A relative neighborhood graph for the Leukemia dataset

2.2 Functionality and GUI

In our toolbox for visualization, navigation and management of the three neighborhood graphs introduced above, which is being developed as a part of the clinical DSS CaseReasoner, we implemented the following functionality:

- node coloring, to represent numeric and nominal attributes;
- node filtering, according to attribute values in the patient record;
- edge coloring and filtering, according to the underlying distance;
- graph (hierarchical) clustering into an arbitrary number of components including a panel for clustering tree navigation on the graph;
- reconfigurable tooltips displaying clinical data from the patient record and images;
- nearest neighbor classification and regression performance visualization for each node, for a selected class attribute and a certain similarity context;
- image visualization within the nodes of the graph (e.g. meshes corresponding to the pulmonary trunk of the patient can be displayed).

Besides clinical data and patient similarities, the neighborhood graphs are nicely suitable for displaying images corresponding to patients. The same operations can still be used as for usual graphs (graph clustering, node coloring and filtering, edge coloring and filtering, etc.); also the images (e.g., meshes) can be scaled and rotated. In Fig. 2 below the meshes corresponding to the pulmonary trunks of the patients are displayed within the nodes of a graph displaying a cohort of HeC cardiac patients, and a sketch of GUI of the neighorhood graph module is shown, including the toolbar with access to the basic operations on the graph such as coloring, filtering and clustering, a pop-up graph settings control panel and a status bar with basic information about the currently displayed graph. The interactive graph navigation is implemented using the *Prefuse* open source data visualization toolkit as the core [Heer *et al.*, 2005].

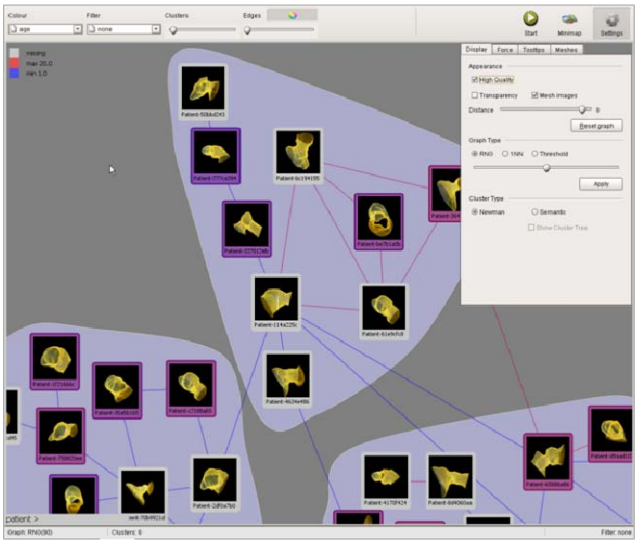


Figure 2 GUI and image visualization within the neighbor graph

The GUI for the basic functionality was designed to be intuitive and straightforward. E.g., if node coloring option is activated, a small legend tells which colors represent the maximum and minimum value or in the case of a nominal value, the range of displayed feature values. In the case of node filtering, the filtered values or ranges are also displayed. Edge coloring represents the distances between the patients with a color range, from blue (weak connection) to red (strong connection). The edge filtering functionality removes a given percent of the weakest connections from the graph, so only the more relevant connections will be remaining.

Besides coloring each node according to a selected attribute, the node color may also be selected to represent the predictive performance of the current similarity context with respect to a certain nominal or numeric feature. In particular, for every node, the leave-one-out estimate of margin or 0/1 loss function with nearest neighbor classification can be displayed, or the leave-one-out estimate of the absolute error with nearest neighbor regression for numeric attributes can be visualized.

As the underlying distance function used for the construction of the neighborhood graph, two options may be considered. The first is to use a certain canonical distance function, such as the well known Euclidean metric, with a possibility to change the influence/importance of every contributing feature. However, due to the inherent complexity of medical data, the canonical distance function may not always represent the true similarity among patients, and thus be suboptimal for decision support. A solution to this issue is to learn a strong distance function for a given classification context under consideration using the available labeled data for training.

Two techniques for learning discriminative distance functions were implemented by us and evaluated; learning from equivalence constraints in the product or difference spaces and the intrinsic Random Forest (RF) distance. Our experiments confirm that both techniques demonstrate competitive performance with respect to the plain learning, and are suitable to be used in combination with the neighborhood graph visualization. The intrinsic RF distance is proven to be more robust overall in our experiments, although finding suitable parameters for learning from equivalence constraints may still be competitive. A thorough introduction to the techniques for learning discriminative distance functions is given in Section 3. We use both the canonical and the discriminative distance functions for constructing the graphs, and in the case of the latter one, customized models can be generated, which can be stored and retrieved later on, in order to specify the similarity context of interest.

For clustering the neighborhood graphs, we use the following two algorithms; (1) the Girvan and Newman's algorithm for graph clustering which is often used for clustering of complex social and biological networks [Girvan and Newman, 2002], (2) top-down induction of a semantic clustering tree (in the original feature space), the goal of which is to provide every cluster with a semantic description that can be inspected by a clinician and may carry important information.

The *semantic clustering* algorithm was developed specifically for CaseReasoner; we could not find a same algorithm already described in the literature, although it is simple and many similar approaches do exist. The related algorithms differ in the structure of the generated cluster descriptions. Our main intention was to provide a *tree* with semantic splits in the nodes that could be used in order to navigate the hierarchical clustering generated and explore the clusters. In order to generate the tree, we use a similar top-down decision tree induction procedure which is often used for supervised learning (e.g. the C4.5 decision tree). Similar to the supervised case, all possible univariate semantic splits are examined in each node (such as '*gender=F*' or '*age<2*'). As the criterion to find the best split, the ratio of between-cluster variance to the within-cluster variance is used. Variance is defined in terms of the current similarity context. If it is specified as a set of features of interest, then the variance can be calculated directly on them. If a customized distance function is loaded, then the variance is represented via distances between a pair of within- and between- cluster cases. According to the first feedback of clinicians regarding the implemented semantic clustering algorithm, the generated tree often contains useful information and may serve as a certain description of the current similarity context.

In Fig. 3 GUI of the neighborhood graph module within the HeC DSS CaseReasoner is shown. The graph shown displays the semantic clustering of a cohort of cardiac patients according to the currently selected similarity context, and node color represents blood pressure for corresponding patient. The pop-up control panel in the upper right corner is used for navigation over the current clustering tree; each cluster can be centered and highlighted and split further by clicking on the corresponding node in the tree in this panel. The panel on the left to the graph navigation panel is the patient record navigation panel which is used in order to browse and compare feature values and their place in the general feature distribution for the current and most similar patient.

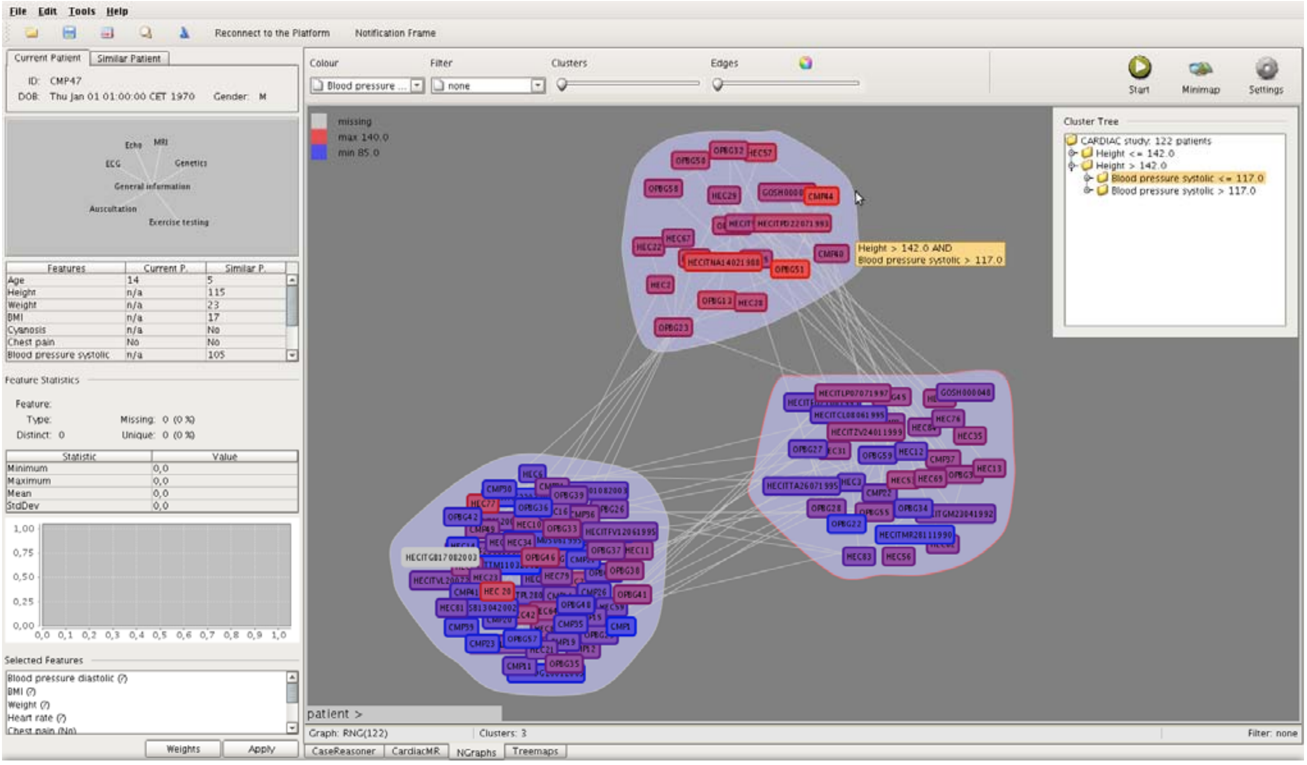
3 Learning Discriminative Distance Functions

There are several reasons that motivate the studies in the area of learning distance functions and their use in practice [Bar-Hillel, 2006]. First, learning a distance function helps to combine the power of strong learners with the transparency of nearest neighbor classification. Moreover, learning a proper distance function was shown to be especially helpful for high-dimensional data with many correlated, weakly relevant and irrelevant features, where most traditional techniques would fail. Also, it is easy to show that choosing an optimal distance function makes classifier learning redundant. Next, learning distance functions breaks the learning process into two sequential steps (distance learning followed by classification or clustering), where each step requires search in a less complex functional space than in the immediate learning. Moreover, it fosters the creation of more modular and thus more flexible systems, supporting component reuse. Another important benefit is the opportunity for inductive transfer between similar tasks; this approach is often used in computer vision applications; see e.g. [Mahamud and Hebert, 2003].

Historically, the most popular approach in distance function learning is Mahalanobis metric learning, which has received considerable research attention but is however often inferior to many non-linear and non-metric distance learning techniques. While distance metrics and kernels are widely used by various powerful algorithms, they work well only in cases where their axioms hold [Hertz, 2006]. For example, in [Jacobs *et al.*, 2000] it was shown that distance functions that are robust to outliers and irrelevant features are non-metric, as they tend to violate the triangular inequality. Human similarity judgements were shown to violate both the symmetry and triangular inequality metric properties. Moreover, a large number of hand-crafted context-specific distance functions suggested in various application domains are far from being metric. Our focus is thus on techniques for learning non-linear and non-metric discriminative distance functions, two important representatives of which are considered in sub-sections below.

More than in any other research domain, the problem of learning a better distance function lies in the core of research in *computer vision* [Bar-Hillel, 2006]. Different imaging applications have been considered, including image retrieval (with facial images, animal images, hand images, and American Sign Language images), object detection (indoor object detection), motion estimation and image registration; see [Hertz, 2006; Bar-Hillel, 2006] for an in-depth review.

Besides vision, some other domains were also considered including computational immunology, analysis of neuronal data, protein fingerprints, and text retrieval [Hertz, 2006]. Surprisingly, there is relatively few related work in text/document retrieval. One example is [Schulz and Joachims, 2003] which studies the retrieval of text documents from the Web by learning a distance metric from comparative constraints.



ID: CMP47
DOB: Thu Jan 01 01:00:00 CET 1970 Gender: M

Features	Current P.	Similar P.
Age	14	5
Height	n/a	115
Weight	n/a	23
BMI	n/a	17
Cyanosis	n/a	No
Chest pain	No	No
Blood pressure systolic	n/a	105

Feature Statistics

Feature:
Type: Missing: 0 (0 %)
Distinct: 0 Unique: 0 (0 %)

Statistic	Value
Minimum	0,0
Maximum	0,0
Mean	0,0
StdDev	0,0

Selected Features

Blood pressure diastolic (?)
BMI (?)
Weight (?)
Heart rate (?)
Chest pain (No)

Colour: Blood pressure ... Filter: none

Cluster Tree
CARDIAC study: 122 patients
Height <= 142.0
Height > 142.0
Blood pressure systolic <= 117.0
Blood pressure systolic > 117.0

Graph: RNG(122) Clusters: 3 Filter: none

Figure 3 GUI of the neighborhood graph module within HeC CaseReasoner

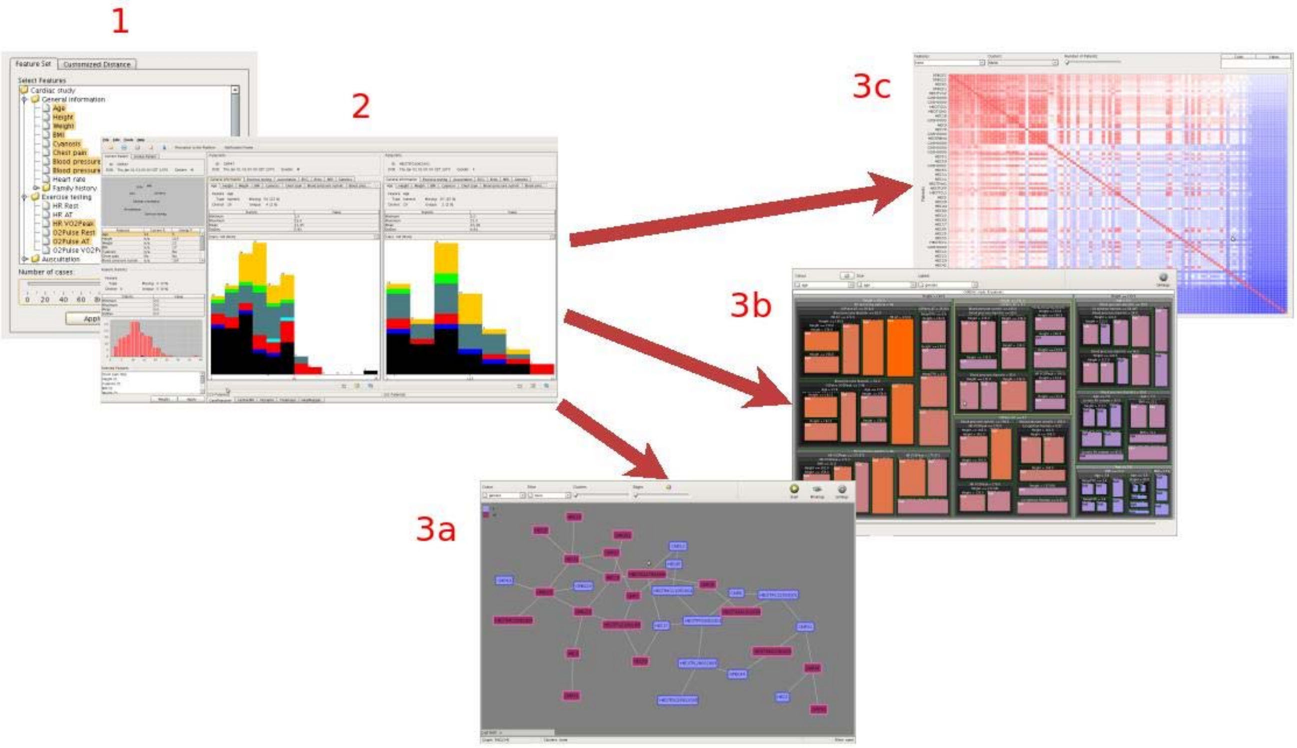


Figure 4 The HeC CaseReasoner application, the workflow

3.1 Learning from Equivalence Constraints

Usually, equivalence constraints are represented using triplets (x_1, x_2, y), where x_1, x_2 are data points in the original space and $y \in \{+1,-1\}$ is a label indicating whether the two points are similar (from the same class) or dissimilar. Learning from these triples is also often called learning in the *product space* (i.e. with pairs of points as input); see [Hertz *et al.*, 2004; Zhou *et al.*, 2006] for examples. While learning in the product space is perhaps a more popular form of learning from equivalence constraints, yet another common alternative is to learn in the *difference space*, the space of vector differences; see [Amores *et al.*, 2006; Yu *et al.*, 2006] for examples. The difference space is normally used with homogeneous high-dimensional data, such as pixel intensities or their PCA coefficients in imaging. While both representations demonstrate promising empirical results in different contexts, there is no understanding which representation is better. No comparison was done so far; usually a single representation for the problem is chosen.

There are two essential reasons that motivate the use of equivalence constraints in learning distance functions; their availability in some learning contexts and the fact that they are a natural input for optimal distance function learning [Bar-Hillel, 2006]. It can be shown that the optimal distance function for classification is of the form $p(y_i \neq y_j | x_i, x_j)$. Under the independence and identical distribution (*i.i.d.)* assumption the optimal distance measure can be expressed in terms of generative models $p(x|y)$ for each class as follows [Mahamud and Hebert, 2003]:

$$p(y_i \neq y_j \mid x_i, x_j) = \sum p(y \mid x_i)(1 - p(y \mid x_j)) \quad (2)$$

3.2 The intrinsic Random Forest distance function

For a Random Forest (RF) learnt for a certain classification problem, the proportion of the trees where two instances appear together in the same leaves can be used as a measure of similarity between them [Breiman, 2001]. For a given forest *f* the similarity between two instances x_1 and x_2 is calculated as follows. The instances are propagated down all *K* trees within *f* and their terminal positions **z** in each of the trees ($\mathbf{z_1}=(z_{11},...,z_{1K})$ for x_1, similarly $\mathbf{z_2}$ for x_2) are recorded. The similarity between the two instances then equals to (*I* is the indicator function):

$$S(x_1, x_2) = \frac{1}{K} \sum I(z_{1i} = z_{2i}) \quad (3)$$

Similarity (2) can be used for different tasks related to the classification problem. Thus, Shi and Horvath [2006] successfully use it for hierarchical clustering of tissue microarray data. First, unlabeled data are expanded with a synthetic class of evenly distributed instances, then a RF is learnt and the intrinsic RF similarities are determined as described above and clustered. The resulting clusters are shown to be clinically more meaningful than the Euclidean distance based clustering with regard to post-operative patient survival.

Interesting is that using this similarity for the most immediate task, nearest neighbor classification, is rather uncommon, comparing to its use for clustering. In one of related works, [Qi *et al.*, 2005], it is used for protein-protein interaction prediction, and the results compare favourably with all previously suggested methods for this task.

The intrinsic RF distance is rather a "dark horse" with respect to learning from equivalence constraints. The number of known applications for it is still limited; perhaps, the most successful application is clustering genetic data, [Shi and Horvath, 2006]. Works on learning equivalence constraints never consider it as a possible alternative. In general, we believe that the circle of applications both for distance learning from equivalence constraints (which is currently applied nearly solely to imaging problems) and for the intrinsic RF distance is still, undeservedly, too narrow and may and should be expanded.

4 CaseReasoner: A Framework for Medical CBR

The basic philosophy behind the design of the CaseReasoner is to provide clinicians with a flexible and interactive tool to enable operations such as data filtering and similarity search over a Grid of clinical centres (following the formerly introduced information retrieval paradigm), and also to facilitate the exploration of the resulting data sets. The aim is to let clinicians explore and compare the patients' records regardless of his/their geographical location, and to visualize their place in the distribution of both the whole population of patients, as well as in the distribution of its semantic subsets.

The selected visualization techniques are implemented to display and navigate through the results of similarity searches in the CaseReasoner. The distance function for similarity search is defined based on a similarity context, which is a subset of features of interest defined by the clinician. The features for each problem domain are organized into a so-called feature ontology, which represents the relationships between features [Tsymbal *et al.*, 2007b]. Similar cases are found both in the whole Integrated Case Database (ICD) over the Grid, and in some subsets of interest (e.g. high-grade tumours, males, a certain node in the Grid, etc), defined in the form of a simple filter. For each patient in the ICD, it is possible to visualize and compare related images from the patient history, thanks to the Gateway's abstracted accesses to backends, storage elements and file catalogs. In combination with the basic feature statistics, class distribution histograms and the scatter plots for the ICD under study, this will be a universal tool for Grid-based decision making in the diseases covered by HeC. Indeed, having a number of clinical centres connected and sharing their data gives the CaseReasoner significant added value. Not only can the CaseReasoner benefit from larger samples but also part of its reasoning logic can be made reusable and delegated to the Grid, with the complexity that it implies.

In short and as illustrated in Figure 4, after selecting a search context (1), the clinician can view basic statistics of the retrieved cases (2) as well as visualize them utilizing neighborhood graphs (3a), treemaps (3b) and heatmaps (3c).

In the development of the general code structure for the CaseReasoner framework we followed a less strict variation of the Presentation-Abstraction-Control pattern (PAC) [Coutaz, 1987]. The main idea behind that pattern is a hierarchical structure of 'agents', where the core of the framework acts as a top level agent, and the modules are subordinate agents. The core and the modules communicate with each other only through their Controller part. This way the flow of data and control remains clear: the general, essential workflow is managed by the core Controller, while the modules can handle the inner business logic of their own. They can request general data manipulating operations through the core, and they are notified about every relevant change as well.

In such a flexible framework any module can be added and removed easily, their development process is fully independent from the whole framework, whose API is well defined and easy to use. Currently CaseReasoner provides five data visualisation modules: the NGraphs module for neighborhood graphs presented above, the TreeMaps, the Heatmapper module, the Patient Panel, to display and compare values and statistics of single patient records, and the CardiacMR module which can visualise and navigate cardiac imaging data related to a selected patient.

5 Discussion

In the evaluation of the HeC platform and in particular the DSS CaseReasoner we follow the “Multi-dimensional In-depth Long-term Case studies” (MILCs) paradigm proposed in [Shneiderman and Plaisant, 2006]. In the MILCs concept the *multi-dimensional* aspect refers to using multiple evaluation techniques including observations, interviews, surveys, as well as automated logging to assess user performance and interface efficacy and utility [Shneiderman and Plaisant, 2006]. In the context of our project, mostly observations, interviews and questionnaires were used so far in order to obtain feedback and assess user satisfaction with the platform. The *in-depth* aspect is the intense engagement of the researchers (the IT experts within Health-e-Child) with the expert users (clinical partners within the project) to the point of becoming a partner or assistant. *Longterm* refers to longitudinal studies that begin with training in use of a specific tool through proficient usage that leads to strategy changes for the expert users. The initial phase of our evaluation has started already in mid 2006, with the start of the project, by demonstrating the preliminary versions of the prototypes for certain platform modules (not yet fully functional) to the clinicians and collecting their requirements to the extension and revision of the tools in so-called knowledge elicitation sessions. The main task of this phase was for the IT partners to better understand the problem domain and the needs of clinicians and develop the tools to fully satisfy these needs. Data collection for the platform has also started in parallel. This iterative evaluation and development phase, when the first fully functional platform prototype was ready, has been gradually replaced (by mid 2008) with the training phase. The main task of this phase which consists of a series of on-site training sessions and will last till the end of the project (April 2010) is to train all the participating clinicians (from the four hospitals in the UK, Italy and France) to use the platform with the collected patient records on the premises of the hospitals and to obtain their extensive feedback in order to better evaluate the platform and fix its discovered deficiencies if needed. Our ultimate goal, which is still for us to achieve is to improve healthcare quality and efficiency and reduce costs for the participating hospitals. *Case studies* refers to the detailed reporting about a small number of individuals working on their own problems, in their normal environment [Shneiderman and Plaisant, 2006].

Perhaps the main competitor to neighborhood graphs as a tool for visualizing patient similarity is heatmaps, which are well known and often used by clinical researchers, in particular by geneticists. In comparison to heatmaps, as follows also from the feedback obtained from partner clinicians in our project, neighborhood graphs possess a number of advantages. In particular, they are easier to read with the more intuitive node-link entity-relationship representation, they allow visualizing additional features or even image thumbnails at nodes, and they have a flexible layout allowing to naturally visualize clusters, enlarge nodes, and filter our a set of nodes and edges.

6 Conclusions

In this paper we introduced a novel technique for visualizing patient similarity, based on neighborhood graphs, which could be helpful in clinical decision making; we also discussed the architecture of our implementation within the Health-e-Child DSS CaseReasoner and in particular the related techniques for learning discriminative distance function. The main advantage of the suggested technique is that the decision support becomes *transparent*; the power of strong machine learning techniques via discriminative distance learning is combined with the transparency of nearest neighbor classification.

An important issue of our ongoing work is a better acquaintance of partner clinicians with the considered neighborhood graph module in the framework of CaseReasoner, and its evaluation in the context of different data classification and decision support tasks. An important issue of our ongoing work with distance learning is the study of the use of online learning techniques for learning from equivalence constraints and in particular the incrementalization of Random Forests, in order to speed up learning in the product space.

Acknowledgements

This work has been partially funded by the EU project Health-e-Child (IST 2004-027749). The authors wish to acknowledge support provided by all the members of the Health-e-Child consortium in the preparation of this paper.

References

[Amores *et al.*, 2006] Jaume Amores, Nicu Sebe, Petia Radeva. Boosting the distance estimation. Application to the *k*-nearest neighbor classifier. *Pattern Recognition Letters* 27, 2006, 201-209.

[Bar-Hillel, 2006] Aharon Bar-Hillel. *Learning from Weak Representations Using Distance Functions and Generative Models*. Ph.D. Thesis, Dept. Comp. Sci., Hebrew Univ. of Jerusalem, 2006.

[Berlin *et al.*, 2006] Amy Berlin, Marco Sorani, Ida Sim. A taxonomic description of computer-based clinical decision support systems. *J. of Biomedical Informatics*, 39(6), 2006, 656-667.

[Breiman, 2001] Leo Breiman. Random Forests. *Machine Learning* 45 (1), 2001, 5-32.

[Clarkson, 2006] Kenneth L. Clarkson. Nearest-neighbor searching and metric space dimensions. (Survey). In: G. Shakhnarovich, T. Darell, P. Indyk (ed.), Nearest-

Neighbor Methods foe Learning and Vision: Theory and Practice, MIT Press, 2006, 15-59.

[Coutaz, 1987] Joëlle Coutaz. PAC: an implementation model for dialog design. In: H-J. Bullinger, B. Shackel (ed.), *Proc. Interact'87 Conference,* North-Holland, 1987, 431-436.

[Heer *et al.*, 2005] Jeffrey Heer, Stuart K. Card, James A. Landay. Prefuse: a toolkit for interactive information visualization. In: *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems, CHI'05*, ACM Press, 2005, 421-430.

[Hertz *et al.*, 2004] Tomer Hertz, Aharon Bar-Hillel, Daphna Weinshall. Learning distance functions for image retrieval. In: *Proc. Int. Conf. on Computer Vision and Pattern Recognition,* 2004.

[Hertz, 2006] Hertz Tomer. *Learning Distance Functions: Algorithms and Applications*. Ph.D. Thesis, Dept. Comp. Sci., Hebrew University of Jerusalem, 2006.

[Jacobs *et al.*, 2000] David W. Jacobs, Daphna Weinshall, Yoram Gdalyahu. Classification with non-metric distances: Image retrieval and class representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(6), 2000, 583-600.

[Jaromczyk and Toussaint, 1992] Jerzy W. Jaromczyk and Godfried T. Toussaint. Relative neighbourhood graphs and their relatives. In: *Proc. IEEE*, 80(9), 1992, 1502-1517.

[Li and Hou, 2004] Ning Li and Jennifer C. Hou. Topology control in heterogeneous wireless networks: problems and solutions. In: *Proc. 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, Infocom'04*, IEEE, 2004.

[Mahamud and Hebert, 2003] Shyjan Mahamud, Martial Hebert. The optimal distance measure for object detection. In: *Proc. Int. Conf. on Computer Vision and Pattern Recognition,* 2003.

[Marcotegui and Beucher, 2005] Beatriz Marcotegui and Serge Beucher. Fast implementation of waterfall based on graphs. In: *Proc. 7th Int. Symposium on Mathematical Morphology, Computational Imaging and Vision*, Vol. 30, Springer, 2005, 177-186.

[Muhlenbach and Rakotomalala, 2002] Fabrice Muhlenbach and Ricco Rakotomalala. Multivariate supervised discretization, a neighbourhood graph approach. In: *Proc. IEEE International Conference on Data Mining, ICDM'02*, IEEE Computer Society, 2002, 314-321.

[Nilsson and Sollenborn, 2004] Markus Nilsson, Mikael Sollenborn. Advancements and trends in medical case-based reasoning: an overview of systems and system development. In: *Proc. 17th Int. FLAIRS Conf. on AI, Special Track on CBR*, AAAI Press, 2004, 178-183.

[Paredes and Chavez, 2005] Rodrigo Paredes, Edgar Chavez. Using the k-nearest neighbor graph for proximity searching in metric spaces. In: Proc. SPIRE'05, LNCS 3772, 2005, 127-138.

[Paredes *et al.*, 2006] Rodrigo Paredes, Edgar Chavez, Karina Figuero, Gonzalo Navarro. Practical construction of k-nearest neighbor graphs in metric spaces. In: 5th Int. Workshop on Experimental Algorithms WEA'06, LNCS 4007, Springer, 2006, 85-97.

[Qi *et al.*, 2005] Yanjun Qi, Judith Klein-Seetharaman, Ziv Bar-Joseph. Random Forest similarity for protein-protein interaction prediction from multiple sources. In: *Proc. Pacific Symp. on Biocomputing*, 2005.

[Scharl and Leisch, 2008] Theresa Scharl and Friedrich Leisch. Visualizing gene clusters using neighborhood graphs in R. Tech. Report 16, Dept. of Statistics, Uni. of Munich, 2008. Available at http://epub.ub.uni-muenchen.de/2110/1/tr016.pdf.

[Schulz and Joachims, 2003] Matthew Schulz, Thorsten Joachims. Learning a distance metric from relative comparisons. *Advances in Neural Information Processing Systems*, NIPS 16, 2003.

[Schmidt and Vorobieva, 2005] Rainer Schmidt and Olga Vorobieva. Adaptation and medical case-based reasoning focusing on endocrine therapy support. In: *Proc. Int. Conf. on AI in Medicine, AIME'05*, LNCS, Vol. 3581, Springer, 2005, 300-309.

[Sebastian and Kimia, 2002] Thomas B. Sebastian, Benjamin B. Kimia. Metric-based shape retrieval in large databases. In: Proc. 16th Int. Conf. on Pattern Recognition, Vol. 3, 291-296.

[Shi and Horvath, 2006] Tao Shi, Steve Horvath. Unsupervised learning with Random Forest predictors. *Computational and Graphical Statistics* 15 (1), 2006, 118-138.

[Shneiderman, 1992] Shneiderman B. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics*, 11(1), 1992, 92-99.

[Shneiderman and Plaisaint, 2006] Ben Shneiderman, Catherine Plaisant. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In: *Proc. AVI Workshop on Beyond time and errors: novel evaluation methods for information visualization, BELIV*, ACM Press, 2006, 1-7.

[Toussaint, 1980] Godfried T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12(4), 1980, 261-268.

[Tsymbal *et al.*, 2007a] Alexey Tsymbal, Martin Huber, Sonja Zillner, Tamas Hauer, Shaohua K. Zhou. Visualizing patient similarity in clinical decision support. In: A. Hinneburg (ed.), *LWA 2007: Lernen - Wissen - Adaption, Workshop Proc.*, Martin-Luther-University Halle-Wittenberg, 2007, 304-311.

[Tsymbal *et al.*, 2007b] Alexey Tsymbal, Sonja Zillner, Martin Huber. Ontology- supported machine learning and decision support in biomedicine. In: *Proc. 4th Workshop on Data Integration in the Life Sciences, DILS'07*, LNBI, Springer, 2007, 156-171.

[Yu *et al.*, 2006] Jie Yu, Jaume Amores, Nicu Sebe, Qi Tian. Toward robust distance metric analysis for similarity estimation. In: *Proc. Int. Conf. on Computer Vision and Pattern Recognition, CVPR, 2006.*

[Zhang and Horvath 2005] Bin Zhang and Steve Horvath. A general framework for weighted gene co-expression network analysis, *Statistical Applications in Genetics and Molecular Biology, 1 (17), 2005.*

[Zhou *et al.*, 2006] Shaohua K. Zhou, Jie Shao, Bogdan Georgescu, Dorin Comaniciu. Boostmotion: boosting a discriminative similarity function for motion estimation. In: *Proc. Int. Conf. on Computer Vision and Pattern Recognition,* 2006.

The TNTBASE System and Validation of XML Documents

Vyacheslav Zholudev
Jacobs University of Bremen
D-28759, Bremen, Germany
v.zholudev@jacobs-university.de

Abstract

TNTBASE is an open-source versioned XML database obtained by integrating Berkeley DB XML into the Subversion Server. The system is intended as a basis for collaborative editing and sharing XML-based documents. It integrates versioning and fragment access needed for fine-granular document content management.

Nowadays well-formedness of electronic documents plays a giant role in the contemporary document workflows and applications have to provide domain-specific validation mechanisms for documents they work with. On another hand, XML is coming of age as a basis for document formats, and even though there are a lot of schema-based validation formats and software available for XML, domain-specific directions still remain unfilled.

In this paper we present the TNTBASE system in general and its validation support for XML documents.

1 Introduction

With the rapid growth of computers and Internet resources the communication between humans became much more efficient. The number of electronic documents and the speed of communication are growing rapidly. We see the development of a deep web (web content stored in Databases) from which the surface Web (what we see in our browsers) is generated. With the merging of XML fragment access techniques (most notably URIs [BLFM98] and XPath [CD99; BBC$^+$07]) and database techniques and the ongoing development of XML-based document formats, we are seeing the beginnings of a deep web of XML documents, where surface documents are assembled, aggregated, validated and mashed up from background information in XML databases by techniques like XQuery [XQu07] and document (fragment) collections are managed by XQuery Update [XQU08].

The Web is constantly changing — it has been estimated that 20% of the surface Web changes daily and 30% monthly [CGM00; FMNW03]. While archiving services like the `Wayback Machine` try to get a grip on this for the surface level, we really need an infrastructure for managing and validating changes in the XML-based deep web.

Unfortunately, support for this has been very frugal. Version Control systems like CVS and Subversion [SVN08] which have transformed collaboration workflows in software engineering are deeply text-based (wrt. diff/patch/merge) and do not integrate well with XML databases and validators for different schema languages for XML, like RelaxNG [Rel] or XML Schema [W3C06]. Some relational databases address temporal aspects [DDL02], but this does not seem to have counterparts in the XML database or XQuery world. Wikis provide simple versioning functionalities, but these are largely hand-crafted into each system's (relational) database design.

In this paper we describe in short the TNTBASE system, an open-source versioned XML database obtained by integrating Berkeley DB XML [Ber09b] into the Subversion Server [SVN08]. The system is intended as an enabling technology that provides a basis for future XML-based document management systems that support collaborative editing and sharing by integrating the enabling technologies of versioning and fragment access needed for fine-granular document content management. Also we discuss our vision of how the validation of XML documents should be done in a way that is conformant to Subversion philosophy and how it fits to the TNTBASE system.

The TNTBASE system is developed in the context of the OMDOC project (Open Mathematical Documents [OMD; Koh06]), an XML-based representation format for the structure of mathematical knowledge and communication. Correspondingly, the development requirements for the TNTBASE come out OMDOC-based applications and their storage needs. We are experimenting with a math search engine [KŞ06], a collaborative community-based reader panta rhei [pan], the semantic wiki SWiM [Lan08], the learning system for mathematics ActiveMath [Act08], and a system for the verification of statements about programs VeriFun [Ver08].

In the next section we will summarize what the TNTBASE system is and what it does. Then we will be ready to cover validation mechanisms (see Section 3) offered by TNTBASE and explain some design decisions. Section 4 will detail ideas for future work regarding validation of OMDoc documents, and Section 5 concludes the paper.

2 The TNTBASE System

2.1 Overview

In this section we provide an overview of TNTBASE to allow a better understanding of Section 3. Details can be found at [ZK09].

A slightly simplified view of the TNTBASE architecture is presented in Figure 1. The core of TNTBASE is the XSVN library developed by the author. The main difference between XSVN and the SVN server it replaces is that the former stores the youngest revisions of XML

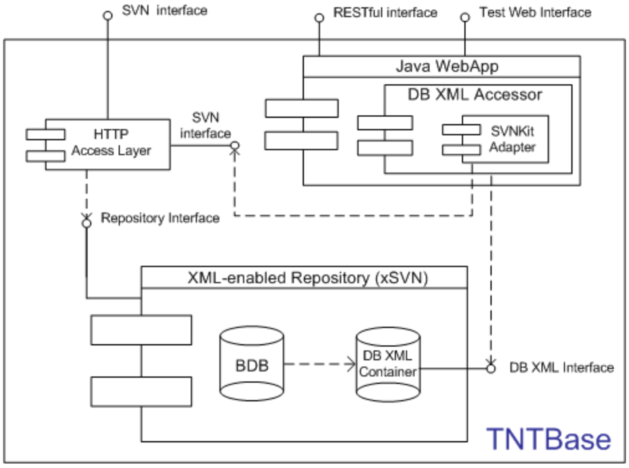


Figure 1: TNTBase architecture

files and also other revisions (on user requests) in Berkeley DB XML (DB XML) instead of Berkeley DB [Ber09a]. This gives us a possibility to employ the DB XML API for querying and modifying XML files via XQuery and XQuery Update facilities. Also such a substitution helps us to keep XML files well-formed and, optionally, conformant to an XML Schema.

In TNTBASE XSVN is managed by Apache's `mod_dav_svn` module or accessed by DB XML ACCESSOR (a Java library which provides a high-level access to DB XML on top of its API) locally on the same machine. Apache's `mod_dav_svn` module exposes an HTTP interface exactly like it is done in SVN. Thereby a TNTBASE user can work with TNTBASE repository exactly in the same way as with a normal SVN repository via HTTP protocol including Apache's SVN authentication via `authz` and `groups` files. In other words any SVN client is able to communicate with TNTBASE. The non-XML content can be managed as well in TNTBASE, but only via an XSVN's HTTP interface.

The DB XML ACCESSOR module can work directly with XML content in an XSVN repository by utilizing the DB XML API. All indispensable information needed for XML-specific tasks is incorporated in a DB XML container using additional documents or metadata fields of documents. SVNKITADAPTER (a Java library which employs SVNKit [SVN07]) comes into play when the revision information needs to be accessed, and acts as a mediator between an XSVN repository and DB XML ACCESSOR. And in turn when DB XML ACCESSOR intends to create a new revision in a XSVN repository it also exploits SVNKITADAPTER functionality.

The DB XML ACCESSOR realizes a number of useful features, but is able to access an XSVN repository only locally. To expose all its functionality to the world TNTBASE provides a RESTful interface, see [ZK09; TNT09b]. We use the Jersey [Jer09] library to implement a RESTful interface in TNTBASE. Jersey is a reference implementation of JAX-RS (JSR 311), the Java API for RESTful Web Services [JSR09] and has simplified our implementation considerably.

TNTBASE provides a test web-form that allow users to play with a subset of the TNTBASE functionality. Also an XML-content browser is available online which shows the TNTBASE file system content including virtual files. United authentication for all interfaces is a subject for a future work.[1]

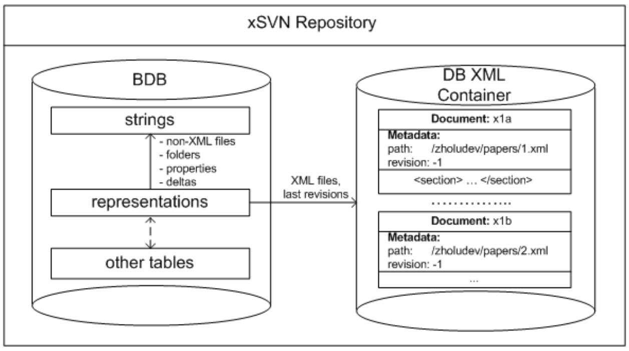


Figure 2: XSVN repository

Currently readers can try out an online TNTBASE test instance (see [TNT09a]), for additional information refer to [TNT09b].

2.2 XSVN, an XML-enabled Repository

Since XSVN is a core of TNTBASE and will be referred to in Section 3, we will cover it here in detail. The architecture of XSVN and thus TNTBASE is motivated by the following observation: Both the SVN server and the DB XML library are based on Berkeley DB (BDB) [Ber09a]. The SVN server uses it to store repository information[2], and DB XML for storing raw bytes of XML and for supporting consistency, recoverability and transactions. Moreover, transactions can be shared between BDB and DB XML. Let us look at the situation in more detail.

The SVN BDB-based file system uses multiple tables to store different repository information like information about locks, revisions, transactions, files, and directories, etc. The two important tables here are *representations* and *strings*. The *strings* table stores only raw bytes and one entry of this table could be any of these:

1. a file's contents or a delta (a difference between two versions of the same entity (directory entry lists, files, property lists) in a special format) that reconstructs file contents
2. a directory entry list in special format called *skel* or a delta that reconstructs a directory entry list skel
3. a property list skel or a delta that reconstructs a property list skel

From looking at a *strings* entry alone there is no way to tell what kind of data it represents; the SVN server uses the *representations* table for this. Its entries are links that address entries in the *strings* table together with information about what kind of *strings* entry it references, and — if it is a delta — what it is a delta against. Note that the SVN server stores only the youngest revision (called the **head revision**) explicitly in the *strings* table. Other revisions of whatever entity (a file, a directory or a property list) are recomputed by recursively applying inverse deltas from the head revision.

To extend SVN to XSVN (an *XML-enabled repository*), we have added the DB XML library to SVN and add a new type of entry in the *representations* table that points to the last version of that document in the DB XML *container*

[1] see Ticket https://trac.mathweb.org/tntbase/ticket/3

[2] In fact SVN can also use a file-system based storage back end (SVN FS), but this does not affect TNTBASE.

(see Figure 2). Containers are entities in DB XML which are used for storing XML documents. Literally, a container is a file on disk that contains all the data associated with your documents, including metadata and indices. For every XSVN repository we use only one container located in the same folder as BDB tables, and therefore it allows us to share the same BDB environment exploited by an SVN back end.

From an end-user perspective there is no difference between SVN and XSVN: all the SVN commands are still available and have the same behavior. But for XML documents the internals are different. Assume that we commit a newly added XML file[3]. Now its content does not go to the *strings* table, but instead a file is added to DB XML container with a name which is equal to the reference key stored in the also newly created *representations* entry of *DB XML full-text* type. Note when we commit a set of files, and even one of XML files is not well-formed then the commit fails and no data are added into an XSVN repository, which conforms to the notion of a transaction in SVN and DB XML. When we want to checkout or update a working copy, XSVN knows what files are stored in DB XML and those files are read from a DB XML container. Another important thing is the scenario when we commit another version of an XML file. The older revision is deleted from DB XML, the newer revision is added to DB XML and a delta between these revisions are stored in the *strings* table. This delta has the normal SVN format and the SVN deltification algorithms have not been changed in XSVN. Thus we are still able to retrieve older revisions of XML documents. Concerning non-XML files the workflow of XSVN is absolutely the same as in SVN: data are stored in the same BDB tables, and the code behaves entirely in the same way. Thereby we are also able to store text or binary data in XSVN which can supplement the collection of XML files (e.g. licensing information or PDFs generated from XML). And moreover we can add or commit XML and non-XML files in the same transaction.

In conclusion: XSVN already offers a versioned XML storage, but without additional modules it is useless as the only difference to SVN is that it refuses to commit ill-formed XML documents. The detailed description of additional services built on top of XSVN is out of scope of this paper (refer to [ZK09] for such information).

3 XML Validation

In this section we will discuss only the XSVN part of TNTBASE and will explain how validation is realized in an SVN-compatible way. Although addition of content is also allowed via RESTful interface of TNTBASE, the most convenient and manageable way of doing this is utilizing an SVN client, and therefore validation should be implemented on the XSVN server and should be managed by any SVN client.

3.1 Why Relax NG?

As we mentioned above, integrating DB XML to the SVN server automatically gives us inspection of XML for well-formedness and conformance to W3C XML Schema associated with a particular XML document. While XML Schema successfully tackled problems with DTDs, it also exposed better opportunities for defining XML languages. But using XML Schema for validation of documents is not always convenient or enough. For instance, an official schemata for OMDOC and MathML [ABC$^+$09] formats are in the Relax NG syntax. Other XML languages may also have no official XML Schema, or a developer of a new XML language may prefer Relax NG. There are a lot of supporters of XML Schema as well as of Relax NG, and there are plenty of disputes about which format is better, but one fact is unquestionable: TNTBASE should support Relax NG validation as well since the language our system focuses on is OMDOC.

There could be two ways of avoiding Relax NG validation in TNTBASE:

1. Make XML Schema as a primary format for describing OMDOC language. Thus the necessity of having Relax NG validation disappears.
2. Use Relax NG as a primary format for OMDOC, but every time OMDOC Relax NG schema is changed, regenerate XML Schema out of it and use the latter in TNTBASE.

The first item does not suit us since XML Schema format has problems that are not presented in Relax NG. Let us enumerate the most significant of them (for more information refer to [XSD09]):

1. XML Schema is hard to read and may be interpreted incorrectly by users not experienced enough.
2. XML Schema provides very weak support for unordered content.
3. XML Schema's support for attributes provides no advance over DTDs.
4. The XML Schema Recommendation is hard to read and understand.

The generation XML Schema out of OMDOC's Relax NG Schema does not work, since even the best converter which the author explored so far — Trang [Tra09] — is not able to convert it. The reason for this is that Trang does not support nested grammars, but they are used in OMDOC's Relax NG.

Thus, the decision was to implement Relax NG validation in TNTBASE.

3.2 Relax NG Validation in XSVN

As was mentioned in the beginning of this section, the validation should be realized in XSVN which is implemented in C/C++ programming languages. The latter fact complicates the integration of many Relax NG validator engines since most of them are written in Java. Most notable of them are: Jing [Jin09] and MSV [MSV09]. The only decent Relax NG validator for C/C++ which the author managed to find so far is Libxml2 [Vei] library. But the serious disadvantage of Libxml2 is its ambiguous and not well-designed error message system. Therefore the decision was to refuse Libxml2 library and make use of one of Java Relax NG validators, namely *Jing*. For integration between C++ and Java *The Java Native Interface (JNI)* [JNI09] has

[3] By default, XSVN considers a file as an XML document if its extension is *.xml* or its *svn:mime-type* property is set to either *text/xml* or *application/xml*. This behavior can be easily adapted, for instance, by checking if a file starts with `<?xml`. Even now an SVN user can benefit from using automated property setting in SVN, i.e. associate certain file extensions with *text/xml svn:mime-type* property. For example, **.xslt* or **.xsd* would obtain *text/xml* mime-type on adding to a working copy and therefore will be treated as XML files for xSVN.

been employed. Due to awkwardness of JNI and, in particular, Java's method invocation from C/C++, the Jing library has been changed in a way that it simplifies Relax NG validation inside XSVN and returns nice-looking error messages back (if any occured). Thus the combination of a modified Jing library and a part of C++ code which invokes methods of this library comprises the XSVN module responsible for Relax NG validation.

3.3 How to Tell XSVN What to Validate?

When we were trying to answer on the question ”How to tell XSVN what to validate”, we wanted to keep an SVN client unchanged and modify only the server side of XSVN. The ultimate solution has two aspects: client and server. On the client side a user has to provide the `tntbase:validate` property for a file he intends to expose for validation. Also this property can be set on a folder recursively, then all files in that folder (and its subfolders) will be validated by XSVN. The value of this property should be a name of a schema. The names of all user-available schemata are stored on a server side in an ad-hoc schema configuration file (SCF) `schemata.xml` which is situated in `db` folder of your repository. The template file is generated automatically during creation of a new repository. An SCF may look like this:

Listing 1: Schemata configuration file

```
<?xml version=”1.0” encoding=”UTF−8”?>
<schemata xmlns=”http://tntbase .mathweb.org/ns”>
    <schema name=”omdoc1.2”
      path=”/home/OMDoc/omdoc−1.2/omdoc.rnc” type=”rnc”/>
    <schema name=”omdoc1.6”
      path=”/home/OMDoc/omdoc.rng”/>
    <schema name=”docbook1.5”
      path=”/vzholudev/Papers/ Balisage / balisage −1−1.rng”/>
</schemata>
```

So as we can see for each name we have a file system path which represents a schema. An XSVN administrator is responsible for setting this up. If a user sets a schema name that is not in an SCF, then the files to be validated against this schema are considered to be invalid and the whole XSVN transaction is aborted. Also if during a commit even one file turned out to be invalid then the whole XSVN transaction is aborted as well, i.e. no files get committed. This perfectly reflects the notion of an SVN transaction. Furthermore, it is not necessary to set up the `tntbase:validate` property right after addition of a file (files). A user can do it at any point of time, e.g. after the revision *21* has been committed. Then after any commit of that file, it will be validated until the `tntnase:validate` property is removed. On Listing 1.1 we can see the use of the attribute `type` that denotes the type of a schema. Currently it could be either `rnc` or `rng` that represents Compact or XML syntaxes of Relax NG respectively. If the `type` attribute is omitted, then the type of Relax NG is calculated depending on the extension of a file in the `path` attribute. If this calculation failed to be done, then validation fails with the corresponding error.

3.4 Versioned Schemata

Often development of documents is accompanied by development of a corresponding schema. Sometimes new types or elements are added or old ones are being evolved in an XML language, and such changes should be reflected in a schema as well. Thus we want to keep a schema in a repository and validate documents against its last version (head revision). In the approach considered in the previous section schemata are meant to be in a file system but not in a repository. It would be relatively easy to change or enhance the format of the `schemata.xml` file in such a way that a schema name points to the path in a repository, and when documents are about to be validated, retrieve the appropriate schema file from a repository. However, things are getting more complicated when the main schema file contains links to secondary schemata. In this case the schema validator — in our case Jing — will not be able to resolve references to other schemata since it does not know where to search for them. There validation will fail even though an arbitrary document is well-formed. One of the solutions would be to implement special entity resolver in Jing which would know how to retrieve schemata by path from a repository.

However, the faster and more elegant solution exists. Assume that we store our schemata in a repository under the path `/main/schemata`. On a server side we checkout this path to a working copy to some place, e.g. `/var/www/schemata`, and update this working copy from a post-commit hook. So our schemata folder is always up-to-date and we can easily link this folder from the `schemata.xml`. The only overhead of this approach is that our data are duplicated: in a repository and in a local file system. The next step would be to place the `schemata.xml` file into repository and allow clients to manage this file remotely, but this is a subject for a future work[4].

3.5 Managing Validation Properties

Let us go back to our XSVN working copies. Setting up a validation property (`tntbase:validate`) for every single file we added might be somewhat cumbersome. Setting a property recursively for the whole directory may reduce our efforts. However, when we add a file later to the directory that has been exposed to the validation property, we do not get such a property for the newly added file. Probably, this behaviour is not what we want to achieve. We want the validation property to adhere to every added file whose parent directory has this property. In general we saw three ways of attaining this that differ in complexity and flexibility:

SVN client approach On the client side leverage the automated property setting which is offered by any SVN client. This method is quite straightforward and has nothing to do with the XSVN server. That is we can associate different validation properties with file extensions. For example, `*.omdoc` files will automatically get `omdoc1.6` validation property on addition to a working copy. For this in the per-user or system-wide configuration area (see Chapter ”Runtime Configuration Area” in SVN Book [CSFP04]) one should modify `config` file by setting property `enable-auto-props` to `yes` and add the following line to the `auto-props` section: `*.omdoc = tntbase:validate omdoc1.6`.

The serious limitation of this method is that we can not set up different validation properties for the files with the same extensions, but which are located in different folders. For instance, if we have two folders for OMDOC documents, one is for version 1.2 and one is

[4] See Ticket `https://trac.mathweb.org/tntbase/ticket/54`

for version 1.6, then we can not associate schemata for different versions of OMDoc for these folders. Moreover, the administration of such a feature is done on the system or user level. That means that automated property setting will be applied for all repositories and all working copies. That might be not desirable in cases when a user works with multiple XSVN repositories that contain documents in different XML languages, but with the same extension, like `*.xml`.

SVN server approach This approach consists in implementing the pre-commit hook which checks the validation property of a parent folder of the committed item, and if the former owns one, then the same validation property is set to the committed item as well. This approach is more flexible than the previous one, but needs additional repository administration efforts (creating and managing hooks). Also it would be impossible to protect a single file in a folder against validation if a validation property has been set on a parent directory.

Combined approach The most scalable solution described here takes advantage of the first two methods. Each file may or may not have a `tntbase:validate` property. If it is presented, then it contains the name of a schema (like we discussed before). If it is not presented, then parent folders are taken into consideration. Each folder also may have a `tntbase:validate` property, but in a different format given by the following BNF: `tntbase:validate ::= (FILE_EXTENSION SCHEMA_NAME)* DEFAULT_SCHEMA_NAME`, i.e. every validation property, if presented, should have the value ext_1 s_1 ext_2 s_2 ... ext_n s_n s_{def}. Thus the files with the extension ext_i are validated against the schema s_i, all other files are validated against schema s_{def}. There is a special reserved schema name `none`, which tells that this file should not be validated. To sum up, when the file f with the extension ext is committed, it is validated against the schema s, where s is determined in the following order:

- If f has a `tntbase:validate` property, then s is extracted from it. Schema name might also be `none`.
- If f does not own a validation property, then the f's extension ext is being searched in the parent folder's validation property. If there is no entry ext s in there, then s is s_{def}. If there is also no default schema name s_{def}, then we repeat this step for the parent folder of the f's parent folder.
- If we achieved the root of a repository and still did not find a schema name for f, then s becomes `none`.

This mechanism is fairly simple and gives an extreme flexibility and scalability. Moreover it does not require further repository administration — everything (apart from defining an SCF) is managed on the client side.

The third method has been implemented in TNTBase as the most sophisticated mechanism for managing `tntbase:validate` properties and defining independent islands of validation.

4 Towards High-Level Format-Specific Validation

We can distinguish tree stages of document validation (most languages exhibit the same problems, but we will use OMDoc language as an example):

1. **XML validation**. Implies well-formedness checking and validity according to a schema (if presented).
2. **Structural validation**. For example, all theorems have proofs, all used symbols are defined and are in scope.
3. **Semantic validation**. On this stage we should check that, for instance, all expressions are well-typed, all proofs are correct, examples contain material about entities they are linked to, etc.

Each stage is stricter than the previous one, and eventually all three should be performed in context of a single system like TNTBase that utilizes auxiliary libraries to achieve a validation goal.

The first stage is already implemented in TNTBase (see Section 3), just expansion of supported schema languages would be a possible task to do in this direction.

Sometimes we need even deeper validation then that is allowable by Relax NG schema (or any other schema language for XML). For example we might be willing to check whether all symbols are visible in an OMDoc document, i.e. each symbol has been defined locally (i.e. in the same file) or has been defined in those documents that are imported in the initial document (and so on recursively). Also we might want to check for redundant or cyclic imports. Consider the following parts of OMDoc documents:

Listing 2: arith1.omdoc

```
<?xml version=”1.0” encoding=”utf−8”?>
<omdoc xml:id=”arith1−omdoc” version=”1.6”
        modules=”CD”
        xmlns:dc=”http: // purl .org/dc/elements /1.1/ ”
        xmlns:cc=”http: // creativecommons.org/ns”
        xmlns=”http: // omdoc.org/ns”>
  ...
  <theory xml:id=”arith1 ”>
    <symbol name=”plus” xml:id=”plus”>
      <metadata>
        ...
      </metadata>
      <type system=”sts.omdoc#sts”>
        ...
        <!−− definition goes here −−>
        ...
      </type>
    </symbol>
      ...
  </theory>
</omdoc>
```

Listing 3: alg1.omdoc

```
<?xml version=”1.0” encoding=”utf−8”?>
<omdoc xml:id=”arith1−omdoc” version=”1.6”
        modules=”CD”
        xmlns:dc=”http: // purl .org/dc/elements /1.1/ ”
        xmlns:cc=”http: // creativecommons.org/ns”
        xmlns=”http: // omdoc.org/ns”>
  ...
  <theory xml:id=”alg1” cdbase=”http: // www.openmath.org/cd”>
    ...
    <imports xml:id=”alg1−imports−arith1” from=”arith1.omdoc#arith1”/>
    ...
    <assertion xml:id=”zero−prop−1” type=”lemma”>
      ...
      <FMP>
        <OMOBJ xmlns=”http://www.openmath.org/OpenMath”>
          ...
            <OMS cd=”arith1” name=”plus”/>
          ...
          </OMOBJ>
        </FMP>
      </assertion>
      ...
  </theory>
</omdoc>
```

On Listing 1.3 we can see the use of the symbol `plus` in the document `alg1.omdoc`. So sticking to our example, we must check that the symbol `plus` is defined in the scope. For this we must check the `imports` statement and then see whether the theory `arith1` of `arith1.omdoc` contains the definition of the `plus`, and it indeed does (see Listing 1.2). Thus if we are able to successfully check all the symbols in `alg1.omdoc`, then we say that this document is structurally valid (in our example). To be precise, we also have to check the absence of cyclic and redundant imports.

Such kind of validation is already available in JOMDoc [JOM] library and is referred to the second stage of document validation (see above). This type of validation provided by JOMDoc should be integrated into TNTBASE as the latter is positioned as an intelligent storage for OMDOC. Currently there is a stub in the XSVN validation engine which allows to have a different values of the `type` attribute in an SCF. So in the future if a schema name is associated with *jomdoc* type of validation then the more intelligent validity check will be performed. This validation usually involves multiple documents to be explored, and therefore the links between those documents should be resolved inside an XML container of XSVN. The special *imports* resolver should be implemented in JOMDoc in order to be able to find entities inside an XSVN's container (but not only in a file system or on Internet) that are referenced via *import* statements of OMDOC language. When this task is completed, we can start integrating the JOMDoc validation mechanism into the XSVN's validation engine. Other types of second stage validation are planned to be incorporated into JOMDoc library (see [Rab08] for more details).

Finally, the third stage is the most complicated one. For more details about this validation could be found at [Rab08]. Currently this is a subject for a future work, and ideas how to accomplish that are not formed clearly enough.

5 Conclusion

We have presented an overview of the TNTBASE system, a versioned XML database system that can act as a storage solution for an XML-based deep web, and discussed the validation mechanisms it exposes or will expose in the future. The implementation effort has reached a state, where the system has enough features to be used in experimental applications. TNTBASE may significantly ease implementation and experimentation of XML-based applications, as it allows to offload the storage layer to a separate system. Moreover users that require only versioning functionality may use TNTBASE as a version control system whereas more exigent users can experiment with additional features of the system. Even those users that need only versioning, can benefit from validating XML documents stored in the repository. That may help to keep a collection of documents more consistent and valid from different perspectives.

Acknowledgments

I would like to thank my supervisor Michael Kohlhase and my colleague Florian Rabe for providing use cases for the system and shooting at me their ideas. Their desire to have a system that totally satisfies their needs caused the considerable evolution of ideas devoted to the management of validation properties.

References

[ABC+09] Ron Ausbrooks, Bert Bos, Olga Caprotti, David Carlisle, Giorgi Chavchanidze, Ananth Coorg, Stéphane Dalmas, Stan Devitt, Sam Dooley, Margaret Hinchcliffe, Patrick Ion, Michael Kohlhase, Azzeddine Lazrek, Dennis Leas, Paul Libbrecht, Manolis Mavrikis, Bruce Miller, Robert Miner, Murray Sargent, Kyle Siegrist, Neil Soiffer, Stephen Watt, and Mohamed Zergaoui. Mathematical Markup Language (MathML) version 3.0. W3C Working Draft of 4. June 2009, World Wide Web Consortium, 2009.

[Act08] ACTIVEMATH, seen September 2008. web page at `http://www.activemath.org/`.

[BBC+07] Anders Berglund, Scott Boag, Don Chamberlin, Mary F. Fernandez, Michael Kay, Jonathan Robie, and Jerome Simeon. XML Path Language (XPath) Version 2.0. W3C recommendation, The World Wide Web Consortium, January 2007.

[Ber09a] Berkeley DB, seen January 2009. available at `http://www.oracle.com/technology/products/berkeley-db/index.html`.

[Ber09b] Berkeley DB XML, seen January 2009. available at `http://www.oracle.com/database/berkeley-db/xml/index.html`.

[BLFM98] Tim Berners-Lee, Roy T. Fielding, and Larry. Masinter. Uniform Resource Identifiers (URI), Generic Syntax. RFC 2717, Internet Engineering Task Force, 1998.

[CD99] James Clark and Steve DeRose. XML Path Language (XPath) Version 1.0. W3C recommendation, The World Wide Web Consortium, November 1999.

[CGM00] J. Cho and H. Garcia-Molina. The evolution of the web and implications for an incremental crawler. In *Proc. of the 26th International Conference on Very Large Databases*, pages 200–209, 2000.

[CSFP04] Ben Collins-Sussman, Brian W. Fitzpatrick, and Michael Pilato. *Version Control With Subversion*. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2004.

[DDL02] C.J. Date, Hugh Darwen, and Nikos Lorentzos. *Temporal Data & the Relational Model*. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, 2002.

[FMNW03] Dennis Fetterly, Mark Manasse, Marc Najork, and Janet Wiener. A large-scale study of the evolution of web pages. In *WWW2003*. ACM Press, 2003.

[Jer09] Reference Implementation for building RESTful Web services, seen April 2009. available at https://jersey.dev.java.net/.

[Jin09] Jing — Relax NG Validator in Java, seen May 2009. available at http://www.thaiopensource.com/relaxng/jing.html.

[JNI09] The Java Native Interface, seen May 2009. available at http://java.sun.com/docs/books/jni/.

[JOM] JOMDoc Project — Java Library for OMDoc documents.

[JSR09] JSR 311: JAX-RS: The Java API for RESTful Web Services, seen April 2009. available at https://jsr311.dev.java.net/nonav/releases/1.0/index.html.

[Koh06] Michael Kohlhase. OMDoc – *An open markup format for mathematical documents [Version 1.2]*. Number 4180 in LNAI. Springer Verlag, 2006.

[KŞ06] Michael Kohlhase and Ioan Şucan. A search engine for mathematical formulae. In Tetsuo Ida, Jacques Calmet, and Dongming Wang, editors, *Proceedings of Artificial Intelligence and Symbolic Computation, AISC'2006*, number 4120 in LNAI, pages 241–253. Springer Verlag, 2006.

[Lan08] Christoph Lange. SWiM: A semantic wiki for mathematical knowledge management. web page at http://kwarc.info/projects/swim/, seen October 2008.

[MSV09] The Sun Multi-Schema XML Validator, seen May 2009. available at https://msv.dev.java.net/.

[OMD] OMDoc. web page at http://omdoc.org.

[pan] The panta rhei Project. seen March 2009.

[Rab08] Florian Rabe. *Representing Logics and Logic Translations*. PhD thesis, Jacobs University Bremen, 2008.

[Rel] A Schema Language for XML. available at http://www.relaxng.org/.

[SVN07] SVNKit - The only pure Java Subversion library in the world!, seen September 2007. available at http://svnkit.com/.

[SVN08] Subversion, seen June 2008. available at http://subversion.tigris.org/.

[TNT09a] TNTBase Demo, seen June 2009. Available at http://alpha.tntbase.mathweb.org:8080/tntbase/lectures/.

[TNT09b] TNTBase Home Page, seen June 2009. Available at https://trac.mathweb.org/tntbase/.

[Tra09] Trang — Multi-format schema converter based on RELAX NG, seen May 2009. available at http://www.thaiopensource.com/relaxng/trang.html.

[Vei] Daniel Veillard. The XML c parser and toolkit of gnome; libxml. System Home page at http://xmlsoft.org.

[Ver08] VeriFun: A verifier for functional programs, seen February 2008. system homepage at http://www.verifun.de/.

[W3C06] XML Schema. http://www.w3.org/XML/Schema, 2006. Seen July 2006.

[XQu07] XQuery: An XML Query Language, seen December 2007. available at http://www.w3.org/TR/xquery/.

[XQU08] XQUpdate: XQuery Update Facility 1.0, seen February 2008. available at http://www.w3.org/TR/xquery-update-10/.

[XSD09] XML Schema vs. RELAX NG, seen May 2009. available at http://www.webreference.com/xml/column59/index.html.

[ZK09] Vyacheslav Zholudev and Michael Kohlhase. TNTBase: a versioned storage for XML. accepted at BALISAGE 2009, available at http://kwarc.info/vzholudev/pubs/balisage.pdf, 2009.

Author Index