Technische Universität Darmstadt
Knowledge Engineering Group
Fachbereich Informatik
Prof. Dr. Johannes Fürnkranz
PD Dr. rer. nat. Ulf Lorenz

Master's Thesis

IT-driven Text-clustering with Respect to Different Thematic Areas

Author: Nikolay Atanasov
Matriculation Number: 1309373
Address: Kirchstrasse 18
64283 Darmstadt
E-Mail-Address: atanasov_nikolay@yahoo.com
Supervisor: Prof. Dr. Johannes Fürnkranz
BMW Supervisor: Dipl.-Ing. Holger Endt
In Cooperation with BMW Group
BMW Research and Technology
Department EG-L-3

Abstract

The goal of this Master's Thesis is to develop an approach for measuring the similarity among documents, stemming from various areas such as scientific literature, belles letters, news, etc. that might be written in different styles and languages. In traditional text clustering methods this is done through the "bag of words" concept. This method calculates the similarity of these documents based on the frequencies of each term found in them, but exhibits the drawback of ignoring the semantic relationship among the words. Consequently, if two documents, representing the same topic use different terms or synonyms, they will be falsely classified as distant.

In order to overcome this problem some external knowledge has to be used. Wikipedia is an appropriate example for a good external dictionary - it is multilingual and written collaboratively by more than 10000 regular editing contributors. Each article describes a single topic. If equivalent concepts, e.g. synonyms or alternative names exist, they are redirected to the same article, section of an article, or page usually from alternative article, describing the main concept. Every Wikipedia page belongs to at least one category. Furthermore, there are links to other languages associated to each topic. Therefore, the information provided by Wikipedia is applied to enhance the "bag of words" technique by proposing an approach that measures the similarity among documents taking into consideration the semantic relationships among the words. For this purpose, all terms of the documents are mapped to the corresponding Wikipedia article. Then the links among them are extracted and used as enrichment to the text representations. As a result, due to the use of additional information, more accurate comparison results are achieved.

Zusammenfassung

Das Ziel dieser Masterthesis ist es, eine Funktion zu entwickeln, die Ähnlichkeiten zwischen Dokumenten auf unterschiedlichen Sprachen, sowie aus verschiedenen Bereichen, wie z.B. Nachrichten, schöngeistige Literatur oder fachliche Texte, ermittelt. In traditionellen Text Clustering Methoden wird dafür das Verfahren “bag of words” verwendet. Die Ähnlichkeit lässt sich durch die Häufigkeit der in diesen Dokumenten gefundenen Begriffe auswerten. Der Nachteil dieser Methode ist jedoch, dass die semantischen Beziehungen zwischen Wörtern ignoriert werden. Wenn beispielsweise zwei Dokumente dasselbe Thema präsentieren aber unterschiedliche Fachwörter oder Synonyme enthalten, können sie fälschlicherweise als fremd klassifiziert werden.

Um dieses Problem umzugehen, muss externes Wissen eingesetzt werden. Dazu bietet sich Wikipedia als ein angemessenes Beispiel für die Benutzung externen Wissens. Die Enzyklopädie ist multilingual und jeder Artikel schreibt über ein bestimmtes Thema. Die Inhalte der Artikel werden regelmäßig durch mehr als 10000 Nutzer verfeinert. Existieren äquivalente Begriffe, wie z.B. Synonyme oder alternative Namen, werden diese zu demselben Artikel weitergeleitet. Jede Wikipedia Seite gehört mindestens zu einer Kategorie und jeder Eintrag wird in vielen anderen Sprachen angeboten. Demzufolge wäre Wikipedia eine zuverlässige Quelle, das “bag of words” Verfahren zu erweitern und zu verbessern, indem alle Fachbegriffe der zu vergleichenden Dokumente mit dem zugehörigen Wikipedia Artikel assoziiert werden. Die hohe Anzahl an Links würde zu der Verfeinerung der Textrepräsentation beitragen. Mit Hilfe dieser zusätzlichen Information ist es möglich sowohl ein besseres als auch ein genaueres Vergleichsergebnis zu erzielen.

Contents

Chapter 1

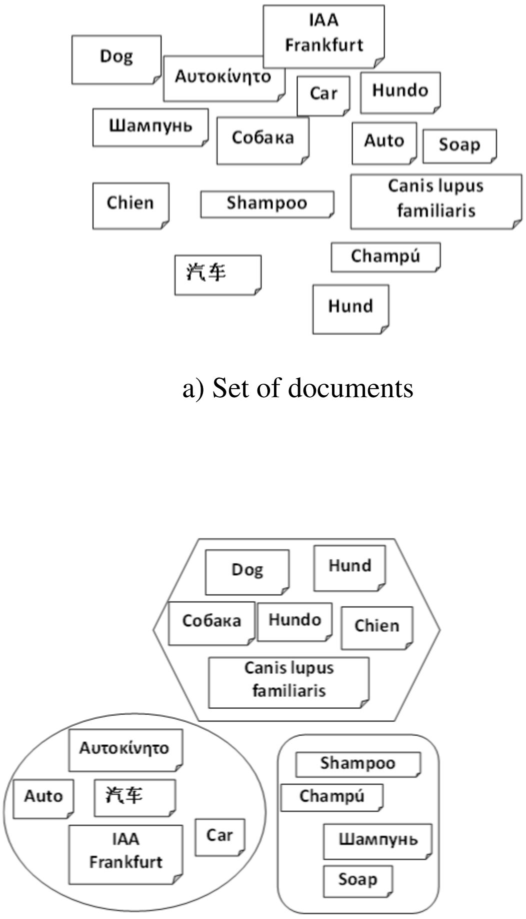
Introduction

1.1 Problem Statement

The amount of articles and documents on the World Wide Web is growing exponentially [15, 12]. Since they are written in various languages, styles and formats, there is a huge need for efficient and fast algorithms in order to ease users' navigation and browsing. The goal of these algorithms is to group the data objects into sets considering the similarity (in terms of topics) among them. A common approach for solving this problem is to apply clustering analysis techniques. This involves dividing the documents into clusters of similar topics, independent of the language in which they are written in. (cf. Figure: 1.1)

The traditional text classification algorithms are based on the "bag of words" (BOW) concept [21, 22]. This approach treats each document as a weighted vector of terms, where the terms are usually mapped to words. However, it ignores the relationships among the terms and considers them only as separate entities. This results in the loss of valuable semantic information. It is exactly the semantic relatedness that defines how related two terms are based on their meaning. For example, the term *engine* is highly connected to the term *automobile*, even thought they do not explain the same object. The semantic connections can also include antonyms, meronyms, hypernyms etc. *Engine* and *automobile* are meronyms, since an engine is part of an automobile. Such information should be taken into account when classifying different data sets. As an example, if two documents use different collections of words to represent the same topic, they can be falsely assigned to different clusters, although the keywords they use are synonyms or semantically associated in other forms. This shows that applying the "bag of words" concept is not sufficient to cope properly with the enormous amount of data that needs to be classified.

The need for more accurate classification of semantically associated words is addressed by enriching the document representation with external knowledge defined by ontologies. Ontology is the structural framework for organizing information. It generally comprises at least three elements: concepts, attributes and the relationships among concepts. To this end, the proposed approach requires finding the ontology concepts for each and every word in the document in order to provide external knowledge base.



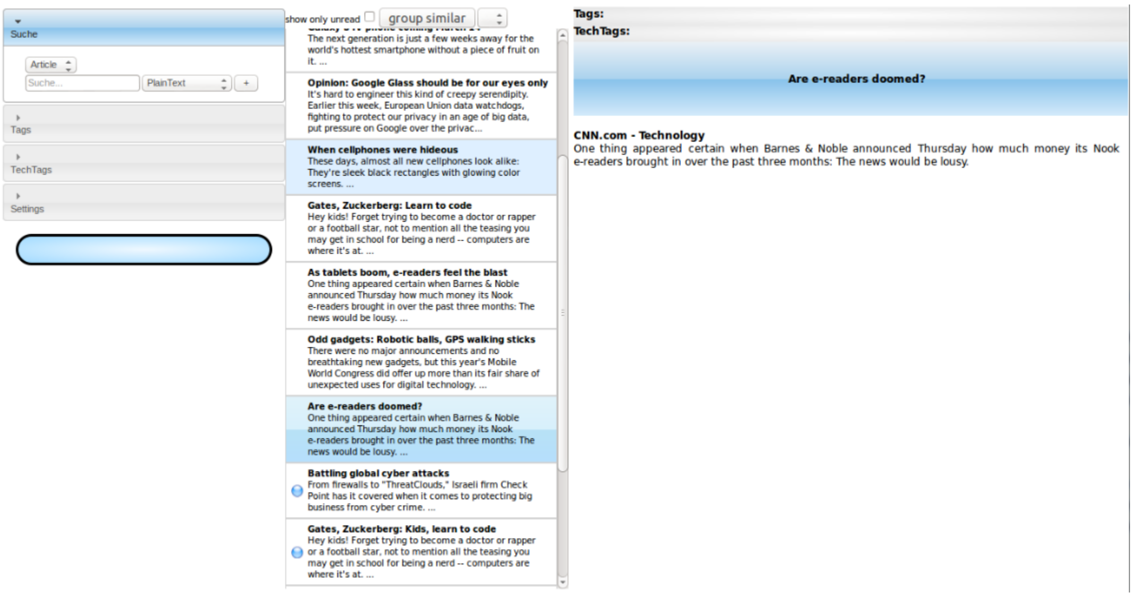
a) Set of documents

b) The document clusters from the set

Figure 1.1: Example of documents clusters.

A typical approach to add external knowledge base to each document is finding the ontology concepts for each and every word in the document. In order to reduce additional data noise, all stop-words should be removed. The newly gained document representation can be used both as a replacement and as an addition of features to the original text.

The main challenge employing this technique is to find an up-to-date extensive database that contains ontologies covering a broad spectrum of terms in a collection of documents. The problem persists especially in the case that the documents are written in different styles and languages e.g., due to containing topics of various areas such as news, scientific/technical literature, etc. In previous works, WordNet [8, 9] has been used as ontology database. WordNet is a lexical database of the English language based on psycholinguistics studies and developed at the University of Princeton. It groups words into sets of synonyms (synsets) and provides short and general definitions. However, it has limited coverage, supporting only one language and is not updated regularly. In this Master's Thesis the structure of Wikipedia is used as an external knowledge database for gaining ontology information.



The next generation is just a few weeks away for the world's hottest smartphone without a piece of fruit on it. ...

Opinion: Google Glass should be for our eyes only
It's hard to engineer this kind of creepy serendipity. Earlier this week, European Union data watchdogs, fighting to protect our privacy in an age of big data, put pressure on Google over the privac...

When cellphones were hideous
These days, almost all new cellphones look alike: They're sleek black rectangles with glowing color screens. ...

Gates, Zuckerberg: Learn to code
Hey kids! Forget trying to become a doctor or rapper or a football star, not to mention all the teasing you may get in school for being a nerd -- computers are where it's at. ...

As tablets boom, e-readers feel the blast
One thing appeared certain when Barnes & Noble announced Thursday how much money its Nook e-readers brought in over the past three months: The news would be lousy. ...

Odd gadgets: Robotic balls, GPS walking sticks
There were no major announcements and no breathtaking new gadgets, but this year's Mobile World Congress did offer up more than its fair share of unexpected uses for digital technology. ...

Are e-readers doomed?
One thing appeared certain when Barnes & Noble announced Thursday how much money its Nook e-readers brought in over the past three months: The news would be lousy. ...

Battling global cyber attacks
From firewalls to "ThreatClouds," Israeli firm Check Point has it covered when it comes to protecting big business from cyber crime. ...

Gates, Zuckerberg: Kids, learn to code
Hey kids! Forget trying to become a doctor or rapper or a football star, not to mention all the teasing you may get in school for being a nerd -- computers are where it's at. ...

Are e-readers doomed?

CNN.com - Technology
One thing appeared certain when Barnes & Noble announced Thursday how much money its Nook e-readers brought in over the past three months: The news would be lousy.

Figure 1.2: Screenshot BRAIN

In contrast to WordNet, Wikipedia is not a structured thesaurus, but it is multilingual, much more comprehensive, up-to-date and very well formatted. These features make Wikipedia a potentially good ontology database for enriching documents' representation and improving multilingual text clustering.

The objective of this Master's Thesis is to develop a methodology based on the structure of Wikipedia for finding a better similarity (or distance) measure between documents written in different languages and styles. The proposed algorithm will be added as a new feature to system created by BMW AG called BRAIN (cf. Figure: 1.2) .

BRAIN is a powerful server-based tool for collecting related items of content, such as RSS feeds, advertisements, press releases, patents and notes. It is equipped with graphical interface for displaying the collected information. The content is gathered in the background or inserted from admin users. It offers "auto-mark as read" function for different documents, grouping them based on user preferences or pre-defined rules. Every user can assign or build their own tags and filters, or subscribe to new RSS feeds. The feature proposed in this Master's Thesis will enhance BRAIN's functionality by providing the users with a mean to retrieve articles on topics considering their preselections, and regardless of language or style the documents are written in.

1.2 Description of the Remaining Chapters

The basic methods for clustering of documents are investigated in Chapters 2 and 3. A more detailed explanation is given about the TFIDF algorithm. Chapter 4 motivates the choice of Wikipedia as a source of external knowledge database, as well as which parts and links of Wikipedia are used. The following two chapters deal with the implementation (Chapter 5) and the experimental results (Chapter 6). In Chapter 7 a parameter setting is proposed. Finally Chapter 8 concludes this work by considering the most important results and providing some ideas on future works.

Chapter 2

Basics

The objective of text clustering methods is to partition an unstructured set of documents into clusters so that:

- The topics of texts within a cluster are very similar
- The topics of documents in different clusters are very different.

In order to deliver good results all clustering algorithms need well defined function to represent:

- a document
- a similarity (or distance) between documents.

All text clustering methods require several steps of preprocessing the documents. The first step is to remove all stop words from the content. A stop word is a word that is considered irrelevant, bringing only noise to the representation of the document, in which it is. For example, *a, the, of, for, with,* and *so on* are often considered as stop words. Words appearing in only one text can also be removed since they do not contribute to the similarity between documents. Furthermore, terms that appear in many documents on the input data can be filtered out as these make almost any text look similar to the others.

After all stop words are removed, the rest words can be stemmed. Stemming is the process in which all words are reduced to their base or root form (stem). The stem has to be identical to the morphological root of the word. It is usually sufficient when related words map to the same stem, even if this stem is not a valid word itself. For example the group of words *fishing, fished, fish* and *fisher* share the common stem form - fish, and can be viewed as its different occurrences.

Having concluded the stemming, the documents are ready to be mathematically modelled. One of the most widely used approaches is representing a document with a vector, capturing the importance of each term in it. Such a representation of an unstructured set of documents in a common vector space is known as the vector space model. If every document d_i is described by the vector $\vec{V}(d_i)$, where each dictionary term has one component, the collection of documents can be viewed as a set of vectors in a vector space, where each term is an axis (cf. Figure: 2.1). Drawback of this form of representation is the losses of relative ordering of the words in each document. For example, the sentence “BMW is faster than Mercedes” will be mapped to the same vector as the sentence “Mercedes is faster than BMW”.

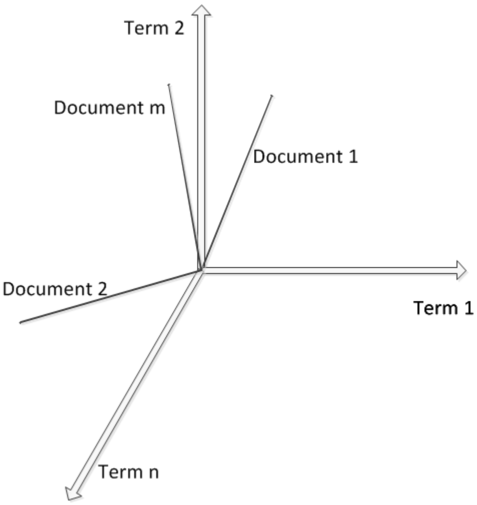


Figure 2.1: Vector Space

Since all documents are represented as vectors, a proper function has to be determined to calculate the similarities between them. The simplest way is to use the magnitude of the vector difference between two text vectors. If such a measure is used, two documents d_1 and d_2 with very similar terms but different in length will be mapped to vectors with huge differences and thus, identified as not related to each other. To overcome this drawback, the cosine similarity (cf. Figure: 2.2) of the document vectors $\vec{V}(d_1)$ and $\vec{V}(d_2)$ is used as:

$$sim(d_1, d_2) = \cos(\theta) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|}. \tag{2.1}$$

The nominator represents the dot product of the two vectors. It is defined as $\vec{x} \cdot \vec{y} = \sum_{i=1}^{n} x_i y_i$, where $x_i = \{x_1, x_2, \ldots, x_n\}$ and $y_i = \{y_1, y_2, \ldots, y_n\}$. The denominator is the product of their Euclidean lengths, which can be computed using:

$$|\vec{V}(d_1)| = \sqrt{\sum_{i=1}^{n} \vec{V}_i^2(d)}. \tag{2.2}$$

The denominator is to length-normalize the vectors $\vec{V}(d_1)$ and $\vec{V}(d_2)$ to unit vectors $\vec{v}(d_1)$ and $\vec{v}(d_2)$, if the vectors are normalized, the cosine similarity can be computed simply by using the dot product. Thus, the problem of finding the documents most similar to a given one, is reduced to finding the text with the highest cosine similarities.

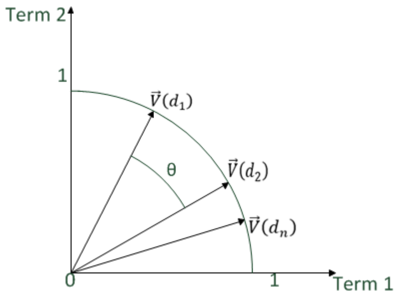


Figure 2.2: Cosine similarity

2.1 Term Weighting

The mathematical representation of the documents can be done in many ways. The simplest one is to use the count of a word in a document as its term weight ($tf(t,d)$), but there are more effective (and more complicated) methods of term weighting. The term frequency $tf(t,d)$ gives an important information of how silent a word is within a document. The higher the term frequency of a word, the more likely it is that it is a good description of the content of the whole document. However, if just the count of the occurrences of a word is used, a document containing one word 5 times will be 5 times more important than a document with only one occurrence of this word. The one with the 5 occurrences should be with higher importance, but not as much as 5 times. That is why, instead of term frequency, another function is calculated like

$$f(t,d) = 1 + \log(tf(t,d)). \tag{2.3}$$

$f(t,d)$ better reflects the importance of a word with more occurrences than only the count of it. For instance, in the previous example the document with 5 occurrences of one term will receive a $f(t,d) = 1,69$ score for this term and the one with only one occurrence will have $f(t,d) = 1$.

The main problem of the function in equation 2.3 is that when it comes to assessing documents from a collection, all words are considered equally important. For example, in a set of texts on movie reviews, almost all documents will have the word movie in them. An improvement to the function should be considered in order to scale down the term weights of terms with high collection frequencies.

With the help of the document frequency function $df(t)$, the proposed function in 2.3 can be enhanced. $df(t)$ shows the number of documents containing the term t, and can be interpreted as an indicator of informativeness. The document frequency can be also scaled logarithmically to $\log \frac{N}{df(t)}$, where N is the total number of documents in the collection. This will give high weight to words that occur in less documents and a zero weight for terms occurring in nearly all documents in the set.

The collection frequency function $cf(t)$, which indicates the total number of occurrences of a word t in a collection, can also be used as an improvement, but the document frequencies give a more reliable measurement[22].

Each document from the collection can be represented mathematically with vectors using the following function for computing the weights for each term.

$$weight(t,d) = \begin{cases} (1+\log f(t,d)) \log \frac{N}{df(t)} & f(t,d) \geq 1 \\ 0 & f(t,d) = 0 \end{cases} \tag{2.4}$$

This form of document frequency weighting represents the *TF-IFD (term frequency-inverse document frequency)* method. It gives to a term *t* a weight in document *d* that is:

1. high if *t* occurs many times in a small number of documents
2. low if *t* occurs fewer times in a document or occurs in many texts
3. nearly 0 if the term occurs in virtually all documents.

Chapter 3

Related Work

To date, most of the research on multilingual text clustering focuses on translating the whole document to an anchor language or on the translation of certain features of the document that describes it best. Considering the first strategy, some authors [4] use machine translation systems, whereas others translate the document word by word, turning to bilingual dictionaries. Furthermore, [4] prove that this method is resource and time consuming, therefore the second strategy is used more often. Other multilingual clustering techniques involve the Latent Semantic Analysis (LSA[1]) based approach [26]. It computes the similarity between terms based on their co-occurrences in different languages. First, the informative terms (nouns and noun phrases) are extracted from each parallel document in the corpus. A term-by-document matrix is created, the rows represent the unique words and columns represent the paragraphs. A mathematical technique (singular value decomposition) is used to reduce the number of columns while preserving the similarity structure among rows. Afterwards, the words are compared by taking the cosine similarity of the angle between the two vectors formed by any two rows. This method only works if the languages are highly related and if they are from the same language family. In the case of languages that are from different families, place or person names are often written (translated) differently. For example, the word “Englisch” in German is written “Anglais” in French.

According to [20] multilingual documents are mapped to a multilingual thesaurus of European languages called Eurovoc[2] to calculate similarities among them. Eurovoc is a multilingual, multidisciplinary ontology managed by the Publication office of the European Union. It contains 24 European languages, making it suitable in the calculation of document similarities in the various European languages. In [20] a method that delivers an automatically generated overview of news is proposed. It works by clustering the multilingual news belonging to the same topic in English, German, French and Italian. Its first step is to extract features from the documents and identify place names. Then, Eurovoc is used to map the documents to a multilingual classification scheme. Through this mapping, a long ranked list of relevant classes for each document is produced. This new language independent representation is used in the calculation of the similarity between multilingual documents.

Most recently, methods have been developed that are not based only on the translation or similarities of terms. For instance, some articles use Wikipedia as external knowledge base for multilingual document clustering. In some of them [13], the basic keyword vector is used to obtain three enriched

[1] LSA is also referred to as Latent Semantic Indexing (LSI)

[2] http://eurovoc.europa.eu/

vectors: Outlink vector, Category vector and Infobox vector. The keyword vector is built after removing all words appearing in more than 50% of the documents and selecting only the top-k words of a document, based on their Term Frequency-Inverse Document Frequency score. As a result, the data noise is reduced and the document's representation is more accurate. For every word in the term vector the corresponding Wikipedia article or its redirect is obtained and - if present - the Outlink-, Category- and Infobox- terms are subsequently extracted and stored in the corresponding enrichment vector. Finally, each document is represented as a linear combination of the four vectors: one basic keyword and the three enriched vectors. Based on the cosine similarity of these combinations, clusters are formed for all languages separately. The centroid of every cluster is calculated by taking the average of all document vectors in this cluster. The clusters with the highest similarity values are merged to form a multilingual cluster. The experiments have showed that the usage of these enriched vectors outperform the baseline. [13] shows, that the most valuable external information is derived from the Outlink vector.

A lot of research has been done on semantic-based classification in a single language. Mostly WordNet has been used for document categorization there. One of the first works [5, 6] to use encyclopedic knowledge for text classification, matches documents with the most relevant Wikipedia articles and then augments the "bag of words" concept using the newly derived information. In [25, 24], the structure of Wikipedia is used as a thesaurus to derive information about the connection between all its concepts: synonymy, hyponymy, polysemy and associative relation. The synonyms and some spelling variations are derived from redirect pages, and the hyponyms can be obtained from the hierarchical structure of Wikipedia (every concept is assigned to at least one category). From the disambiguation pages we can have all the polysemy of an article and associative relations can be received with the help of all the hyperlinks in-between Wikipedia articles. These additional enrichments are used in the calculation of the similarity measure between two documents:

$$S_{Overal} = \lambda_1 \cdot S_{TFIDF} + \lambda_2 \cdot S_{olc} + (1 - \lambda_1 - \lambda_2) \cdot Dis_{Category}. \tag{3.1}$$

Where S_{olc} is calculated by computing the cosine similarity of the out-link categories vectors of two articles. The distanced based measure is computed by the following formula:

$$Dis_{Category} = \frac{length(c_1, c_2)}{D}, \tag{3.2}$$

$length(c_1, c_2)$ is the number of nodes along the shortest path between the categories of the two articles and D is the maximum depth of the taxonomy in which the articles are found. After tuning the parameters λ_1 and λ_2, the best results have been received with $\lambda_1 = 0.4$ and $\lambda_2 = 0.5$. The experiments have delivered a good improvement over the baseline categorization.

Chapter 4

Approach

4.1 Reasons for Using Wikipedia

Wikipedia (founded in 2001) has become the world's largest free online encyclopedia with millions of articles edited collaboratively by volunteers. There are 262 different language versions of the website, although the English, German and French versions have the most articles. The accuracy of Wikipedia is comparable to the one of the Encyclopedia Britannica [7, 23]. The information is very up to date because of the continuous contribution of users. Moreover, Wikipedia is well-formed. Each article describes a single topic. Its title is a succinct phrase resembling a term in a conventional thesaurus. An article must belong to at least one category and can have many internal and external links.

As an open source project, Wikipedia creates and releases periodically dumps of its entire content.[1]

4.1.1 Wikipedia as an External Knowledge Base for Multilingual Document Clustering

Using Wikipedia as an external knowledge base has many advantages compared to other existing resources for multilingual document clustering. Wikipedia is a very dynamic and quickly growing resource, which can be easily transformed and used as a comprehensive and contemporary thesaurus. Many relations between the different articles (terms) can be discovered from the structure of Wikipedia. Through these relations synonyms, hyponyms and other semantic connections can be discovered.

4.1.2 Redirects as Synonyms

Each topic is described by only one article, so if a synonym of a topic is being searched for, a redirect page will be returned. A redirect page, containing only a link to the preferred name in Wikipedia exists for every alternative name of a topic. For instance, if "car" is searched, we receive a redirect to the article "Automobile". The redirects also handle capitalization, spelling variations, abbreviations, scientific terms and colloquialisms. Thus, this is a good way to fight some common misspellings.

[1] http://dumps.wikimedia.org/backup-index.html

For example, the article describing the United States can be found with the search names: “USA”, "U.S.A" (acronyms), "United states of America", “Untied States” (misspelling) and even “Yankee land” (synonym).

4.1.3 Disambiguation Pages

Wikipedia also contains a lot of disambiguation pages, created for the ambiguous terms. An ambiguous term is a word with multiple definitions and multiple Wikipedia articles. For instance, “plane” is a term which can mean an airplane, or a plane in geometry, or a clipping plane (3D computer graphics term). Wikipedia delivers 21 possible articles for this term. With the help of the disambiguation pages, all possible meanings of an ambiguous term will be listed and the user could select the exact article of the intended term.

4.1.4 Categories as Hyponyms

In Wikipedia, both articles and categories must belong to at least one category. This category structure can be used to obtain an important “is a” relation or hyponym of the searched term. A hyponym is a word whose semantic field is included within that of another word. (cf. Figure: 4.1) The article for “automobile” belongs to two categories: “Automobiles” and “Wheeled vehicles”, and “Automobiles” is a subcategory of “Transport” and “Vehicles”.

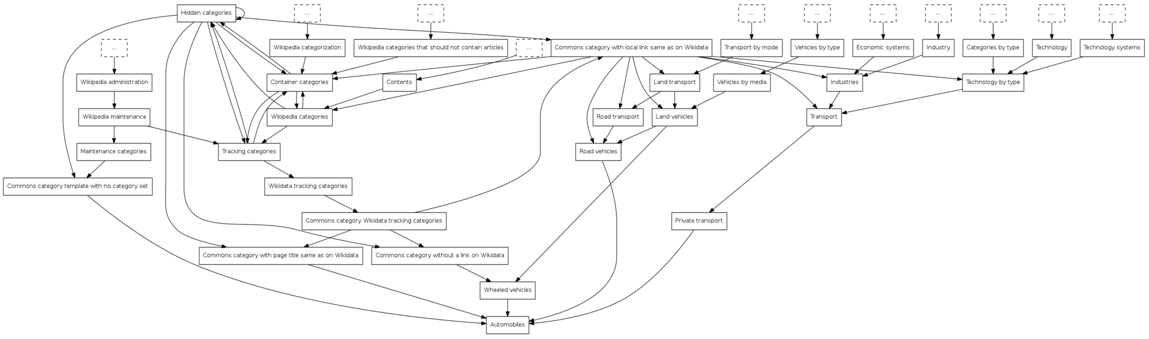


Figure 4.1: Categories of Automobiles

4.1.5 Hyperlinks as Semantic Connections

Every article in Wikipedia can contain a lot of external and internal links to other articles or web pages expressing the associative relatedness between the different articles. For example from an article about "smartphone" links to articles about "Android", "iOS", "Symbian", "BlackBerry OS" and many others are given.

4.1.6 Cross-lingual Links as Dictionary

One of the most valuable links, for the multilingual clustering, are the inter-language ones. Every article is linked to the corresponding one in different languages: the English article "USA" is linked to the German one "Vereinigte Staaten", the Spanish "Estados Unidos" article, the Russian one "Соединённые Штаты Америки" and many others.

4.2 Methodology

As mentioned before, the "bag of words" (BOW) approach only manages the terms explicitly mentioned in the text documents, thus it leaves semantic relationships between important terms. This is the reason why external knowledge should be added to the text representation. Wikipedia will be used as a source or thesaurus to enrich the BOW approach.

4.2.1 Text Representation

Let $D = \{d_1, \ldots, d_n\}$ be a collection of documents d_i and $T = \{t_1, \ldots, t_m\}$ be the set of all different terms occurring in D. After removing all stop words, with the help of a standard stop words list, and stemming the rest, using the well-known Snowball algorithm[2], a term vector representation $\vec{t_d}$ is constructed for each document from the collection (cf. Algorithm: 4.1) . The term vectors are denoted as $\vec{t_d} = (tf(t_1, d), \ldots, tf(t_m, d))$. Next TFIDF weighting for each term is applied and the term vector $\vec{t_d}$ is replaced by $\vec{t_d} = (TFIDF(t_1, d), \ldots, TFIDF(t_m, d))$.

[2] http://snowball.tartarus.org/

Algorithm 4.1 Remove stop words, stem and calculate TFIDF.

```
Initialize uniqueTermsList
Initialize wikipediaArticles

FOR each document of the collection
        FOR each term of the document
                IF term isn't stopWord
                        IF uniqueTermsList doesn't contain term
                                ADD term to uniqueTermsList
                        END IF
                        ADD Stemmed(term) to document.
                           ListOfStemmedTerms
                END IF
        END FOR
END FOR

FOR each term in uniqueTermsList
        Extract from local database corresponding wikipedia article
            and the needed links
        Store the extracted articles in wikipediaArticles
END FOR

FOR each document of the collection
        Calcualte TFIDF score of all terms from document.
           ListOfStemmedTerms
END FOR
```

4.2.2 Text Enrichment

With all term vectors calculated, the document enrichment with external knowledge can start. Thus the main problem of BOW can be solved. The synonyms are found and more general and associative concepts, helping in the identification of related topics, are introduced. For example, in a document, containing the term “airplane” relations with “Boeing”, “Airbus” and many others will be added. Six Wikipedia enrichments are proposed and tested in this Master’s Thesis. Therefore, additional six vectors are created for every document in the collection.

4.2.2.1 Category Based Similarity Measure

If two or more documents have many overlapping Wikipedia categories of their terms, they should be assigned to the same cluster. For instance, if a document contains “BMW” and another one has “Mercedes-Benz” in it and the categories of these two terms are taken from Wikipedia, there is 75% coincidence in them. (cf. Figure: 4.2) That is why a category vector $\overrightarrow{cat}(d)$ is built. It contains the corresponding Wikipedia category Ids for each term in the document.

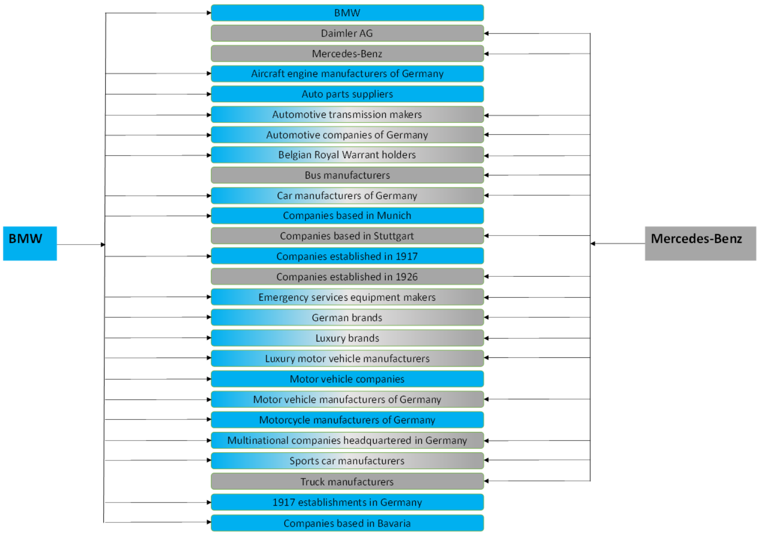

BMW
Mercedes-Benz
BMW
Daimler AG
Mercedes-Benz
Aircraft engine manufacturers of Germany
Auto parts suppliers
Automotive transmission makers
Automotive companies of Germany
Belgian Royal Warrant holders
Bus manufacturers
Car manufacturers of Germany
Companies based in Munich
Companies based in Stuttgart
Companies established in 1917
Companies established in 1926
Emergency services equipment makers
German brands
Luxury brands
Luxury motor vehicle manufacturers
Motor vehicle companies
Motor vehicle manufacturers of Germany
Motorcycle manufacturers of Germany
Multinational companies headquartered in Germany
Sports car manufacturers
Truck manufacturers
1917 establishments in Germany
Companies based in Bavaria

Figure 4.2: The Wikipedia Categories of BMW and Mercedes-Benz

4.2.2.2 External Links Based Similarity Measure

This similarity measure is based on the external links of the articles in Wikipedia. A new vector $\overrightarrow{extLink}(d)$ is generated for every document in the collection.

4.2.2.3 Wikipedia Ids Based Similarity Measure

Every article in Wikipedia has a unique Id number. Thus, the Id vector $\overrightarrow{Id}(d)$ is built, where the Ids of all articles for the terms in a document are shared. This vector resolves the problem with all synonyms and spelling variations of the terms in a document mentioned in Section 4.1.2

4.2.2.4 Internal Links Based Similarity Measure

This vector $\overrightarrow{InLink}(d)$ contains the Ids of all internal links for the Wikipedia articles describing a document (cf. Figure: 4.3). The larger the number of shared internal Link Ids, the stronger the relation between the two documents.

4.2.2.5 Internal Links Categories Based Similarity Measure

With this measure the categories of all internal links of an article are compared. Internal link categories of an article are the categories to which internal link article to the original one belong (cf. Figure: 4.3). So, for each document a vector $\overrightarrow{inLinkCat}(d)$ is created.

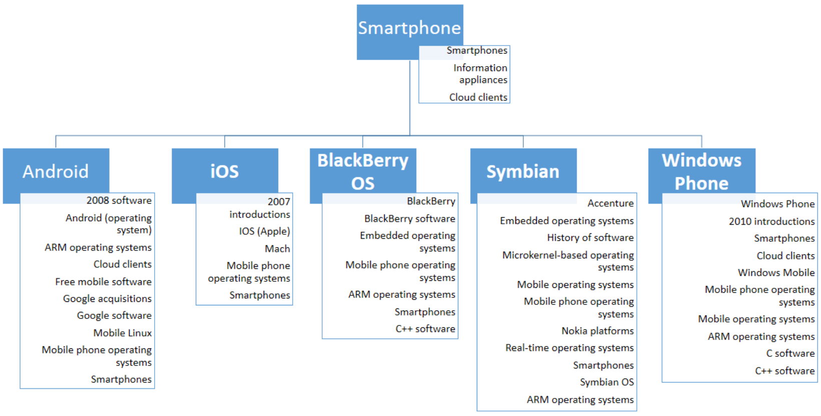


Figure 4.3: Internal links and categories of the Smartphone article

4.2.2.6 Crosslingual Vector

The vector $\overrightarrow{crossLing}(d)$ is created only for the documents which are not in the base language. The base language should be chosen with respect to the number of documents written in this language and also to the size of articles in Wikipedia for the language. The crosslingual vector stores every term describing the document in the base language. With the help of this vector, documents from different languages can be clustered together without many difficulties by building the rest text enrichment vectors only for the base language. For example, if English is the base language and a document written in German is going to be processed, the crosslingual vector, containing the English translation of the German terms, is created first. Then the other five enrichment vectors are built with components taken from the English Wikipedia using the entities of the crosslingual vector. In this way, the clustering can be done straightforward in one language, with these five vectors.

4.3 Calculating the Similarity Between Two Documents

Having TFIDF and enrichment vectors extracted, the similarities between documents can be finally measured by a linear combination among the vectors. This is done with the help of the following equation:

$$\begin{aligned} sim(d_n, d_m) \; = \; & \alpha \cdot sim(d_n, d_m)^{TFIDF} + \beta \cdot sim(d_n, d_m)^{extLinks} + \\ & \gamma \cdot sim(d_n, d_m)^{Ids} + \delta \cdot sim(d_n, d_m)^{Cat} + \\ & \eta \cdot sim(d_n, d_m)^{inLinks} + \vartheta \cdot sim(d_n, d_m)^{inLinksCat}. \end{aligned} \tag{4.1}$$

Where $sim(d_n, d_m)^{TFIDF}$is the cosine similarity of the TFIDF scores of the two documents and the other $(sim(d_n, d_m)^{extLinks}$, $sim(d_n, d_m)^{Ids}$, $sim(d_n, d_m)^{Cat}$, $sim(d_n, d_m)^{inLinks}$ and $sim(d_n, d_m)^{inLinkCat})$ are the corresponding cosine similarity results from the Wikipedia enrichments. The weight parameters control the influence of the similarity measures. A reasonable optimization is proposed in Chapter 7.

Chapter 5

Implementation

5.1 Importing Wikipedia Dumps

As mentioned before, Wikipedia contains a record of table structure and the data from its database, that is stored in dumps, is offered to be downloaded for free by interested users[1]. These dumps are created periodically, at the very least monthly, and usually twice a month. The snapshots contain complete copy of all Wikipedia articles, in a form of text source and metadata embeded in XML. Furthermore, they also include the database tables available in SQL form. These tables hold information for all Wikipedia articles like redirects, external links, their categories and their translation in the other languages. (cf. Figure: 5.1) For the purpose of this Master's Thesis only the links and redirects are needed, therefore the XML files are not taken into consideration.

The following tables are imported in the local database:

- CATEGORY - contains all categories with their internal Id
- CATEGORYLINKS - stores the category names for all Wikipedia articles
- EXTERNALLINKS - all external links for the articles are saved there
- LANGLINKS - stores the translation of all Wikipedia pages in all languages
- PAGE - the main table in the database, there all article titles and identifications are saved
- PAGELINKS - all internal links are stored in this table
- REDIRECT - contains the main (target) article title for each redirect article.

Since most of the tables have a size larger than 1 GB, some tuning and optimizations must be made in order to speed up their import to the local database. As the dumps are not wrapped as a transaction, the database reindexes after each individual insert, which slows the operation crucially. As a result, two scripts are applied before and after each table import.

In this Master's Thesis MySQL is used and by default it runs with autocommit, unique_checks, foreign_key_checks modes enabled. When autocommit is on, the execution of a statement, that updates

[1] http://dumps.wikimedia.org/backup-index.html

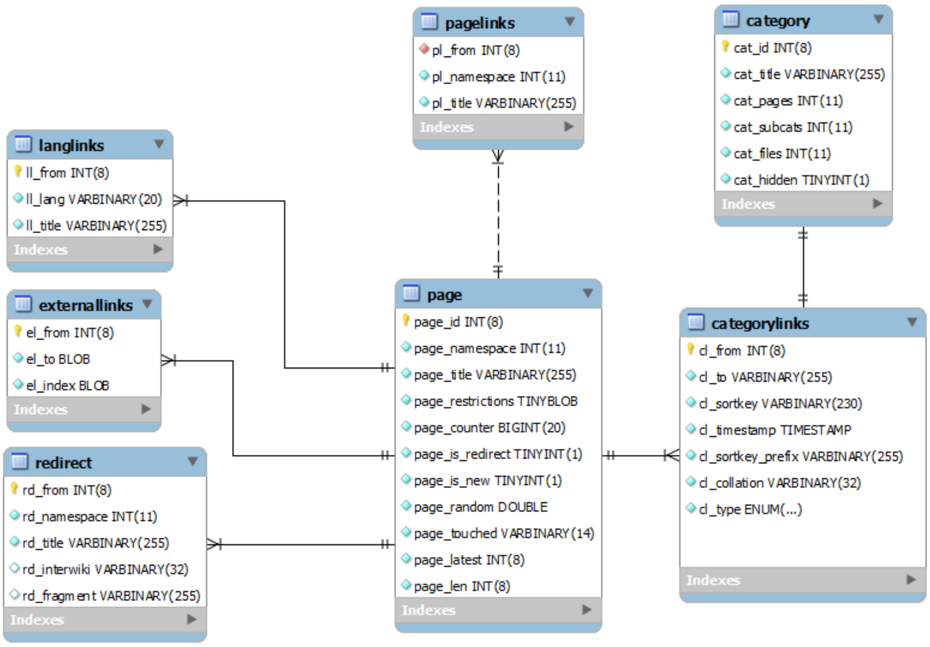


Figure 5.1: Wikipedia tables schema

or modifies a table, is immediately saved on the hard drive to make it permanent, costing many insufficient I/Os. Since the dumps are directly exported from the Wikipedia database, guaranteeing the absence of duplicate keys, checks for uniqueness can be ignored. Disabling the foreign key checks allows insertion into tables in an arbitrary order, different from that required by their parent/child relationships.

Listing 5.1: The Preimport script

```
SET autocommit=0;
SET unique_checks=0;
SET foreign_key_checks=0;
BEGIN;
```

The Preimport (cf. Listing: 5.1) script prevents MySQL from calculating indexes until the entire data set has been read. After the import of a table is successfully executed the exact opposite script (cf. Listing: 5.2) is run.

Listing 5.2: The Postimport script

```
COMMIT;
SET autocommit=1;
SET unique_checks=1;
SET foreign_key_checks=1;
```

In addition, some changes have to be made in the database configuration files. In MySQL such an optimization is done mainly through the my.cnf file. This file stores default startup options for both the server and the clients. Changing the following variables speed up the import of large tables significantly:

- innodb_buffer_pool_size: This and the next variable are the most important, if Innodb tables are used. The default value for the buffer pool size is 8MB, which is totally inefficient, so it must be set to 70-80% of the available memory. The larger the value, the less I/O is needed to access data in tables.
- innodb_log_file_size: Again, if a larger value is set, less checkpoint flush activities are needed in the buffer pool, saving disk I/O. But larger log files mean the recovery is slower in case of crash. So a good balance between a reasonable recovery time and good performance must be found. In the following experiments, the log file size is set to 1512MB.
- innodb_log_buffer_size: This sets the size of the buffer that InnoDB uses to write to log files on disk. A large log buffer enables large transactions to run without a need to write the log to disk before the transactions commit. Thus increasing the log buffer size saves disk I/O.

The whole my.cnf file can be found in the Appendix A.

5.2 Implementation

This chapter explains in details how similarities among documents are calculated. With the initialization of BRAIN, all documents from its database are extracted and kept in a Dictionary with their hash as a key and their plain text as a value. The algorithm, developed in this Master's Thesis, starts when an article in BRAIN is selected and the "Show Similar" button is pressed. (cf. Figure: Figure:5.2) By pressing the button a calculation of different text representations is triggered.

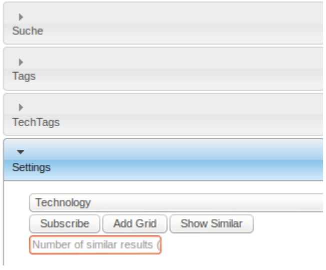


Figure 5.2: Settings in BRAIN

Firstly, all stop words are removed from the texts. After stemming the rest words, the TFIDF representations of the documents, written in the same language as the selected article, are calculated. Then, the enrichment with external knowledge delivered from Wikipedia is performed. A term vector, containing the unique words from all documents as entries, is created. Simultaneously, two dictionaries *wikiArticlesIds* and *wikiArticles* are initialized. The first one has a *string* as a key and *int* as a value. It stores all words with their corresponding Wikipedia article Ids. If no article is found for the given term, its id is set to -111. The second dictionary has *int* as a key and an object *WikipediaArticle* (cf. Listing: 5.3) as a value. These represent the Id of a Wikipedia page and the information of this article respectively.

Listing 5.3: WikipediaArticle

```
public class WikipediaArticle  {
  DatabaseConnector database;
  public int Id {get; set;}
  public bool ambiguous {get; set;}
  public List<string> externalLinks {get; set;}
  public List<int> internalLinksCategories{get; set;}
  public List<int> internalLinks{get; set;}
  public List<int> categoryIds{get; set;}
  Dictionary<int, string> languageLinks{ get; set; }
  public WikipediaArticle(string word,
        DatabaseConnector database ,int language){
  }
}
```

The class WikipediaArticle has a main constructor taking a string word, a database connector and an integer value, representing the language of the word. When the object is initialized, all of its variables are set. At the beginning, the Wikipedia article Id is taken from the database and is checked whether it is a redirect or an ambiguous page. This is done by *getWikiArtickelId*, such an information is obtained from the PAGE table with the help of the query in Listing 5.4

Listing 5.4: Wikipedia Article Query

```
SELECT * FROM page
WHERE page_namespace = '0'AND page_title = "...";
```

page_title represents the searched word. *page_namespace* defines the intended use of the article. For instance, it can be a user profile, help page, template page, talk, etc. Since real content articles are needed, *page_namespace* is set to 0 and is added as an additional restrictive condition to the query in Listing 5.4. If the received article is a redirect, i.e. the word is a synonym, spelling variation or abbreviation, the main Wikipedia page for this word has to be found. The procedure *getRedirectId* returns the Id of the searched redirect. It joins the tables PAGE and REDIRECT on article_title. The main Wikipedia article Id is returned, as seen in query from Listing 5.5

Listing 5.5: Redirect Id Query

```
SELECT b.page_id FROM redirect a
        INNER JOIN page b ON a.rd_title = b.page_title AND b.
           page_namespace = 0
WHERE a.rd_from='...';
```

If the Wikipedia page is not a redirect, then an ambiguous check is conducted. This is done by the function *checkIfAmbiguousPage*. It verifies whether the category “disambiguation_pages” is among the categories of the given article. If the function returns true value, a flag is set. After the checks, all variables of the object are noted. From the table LANGLINKS all translations are acquired. EXTERNALLINKS and PAGELINKS provide all external and internal links respectively. To obtain the categories of an article, the method *getWikiCategoryIds* is executed. It combines the tables CATEGORY and CATEGORYLINKS and delivers the Ids of all categories. Furthermore, some of the internal Wikipedia categories, such as "Hidden_categories", “Good_articles”, “Articles_with_French_language_external_links”, “clean_up_articles “, etc., are removed as they do not contribute to the similarities between articles. To get the categories of the internal links, the function *getInternalLinksCategories* is called. This function follows a specific set of steps. Firstly, a string with all Ids of the internal links is generated and then added to the query in Listing 5.6

Listing 5.6: Internal Links Categories Query

```
SELECT a.cl_to, b.cat_id FROM categorylinks a
        INNER JOIN category b ON a.cl_to = b.cat_title
WHERE a.cl_from IN (...);
```

This query is then submitted to the database and, as a result, the Ids of all categories for the internal links are retrieved.

If the collection of documents contains text in different languages, the Wikipedia translation to a chosen main language of all terms is extracted. This is done in the *getArticleNameInOtherLanguage* method. It takes the article Id and the desired language as an input and returns the name of the page in this language.

After the relevant information for all unique words is extracted from Wikipedia and stored in these two dictionaries, the similarities among the different documents are calculated. Firstly, an object *Document* is created for each text from the collection of documents. Since a vector, with dimension equal to the number of the unique words is needed for every document, the calculating costs expand tremendously. In order to keep the increased complexity manageable, *wikipediaIdsCount, wikipediaExtLinksCount, wikipediaCategoriesCount, wikipediaInternalLinksCount, wikipediaInternalLinkCategoriesCount* and *wikipediaAmbiguousArticleCoun*t dictionaries are developed in every object. They retain the corresponding Wikipedia information for all non stop words of a document. Each dictionary has the acquired Wikipedia information as a key and the number of its occurrence as a value. For instance, a document containing the terms “automobile” and “car” at the same time, will have the Id, categories and links of the Wikipedia article “Automobile” with an occurrence at least 2, because these two terms are redirected to the same article.

After creating objects for all documents and filling them with the needed information, the similarity measures are computed. This is done by an object called “*WikipediaSimilarity*”. (cf. Listing: 5.7) It gets two documents as an input and has five methods that determine the corresponding Wikipedia similarity measures between these two documents. In each of these methods two vectors of the same dimension are created. The vectors are the corresponding Wikipedia representation of the documents. For example, if one of the documents contains 4 Wikipedia articles with category “Vehicles” and the other has none within these category, the Wikipedia category vector for the first document will have 4 as an entry whereas the other will have 0.

Listing 5.7: WikipediaSimilarity Object

```
public class WikipediaSimilarity {
        Document doc1, doc2;
        public WikipediaSimilarity (Document doc1, Document doc2)
                {... }
        public double compareIds()
                {...}
        public double compareExternalLinks()
                {...}
        public double compareInternalLinks()
                {...}
        public double compareCategories()
                {...}
        public double compareInternalLinksCategories()
                {...}
}
```

Chapter 6

Experimental Results

6.1 Wikipedia Data

In this Master's Thesis the English dumps release on March 4, 2013 and the German dumps made public on the June 2, 2013 are used. After importing the dumps, two databases with total size of over 150 GB are obtained. (cf. Figure: 6.1) The English Wikipedia has more than 4 million unique articles and more than 30 million redirect pages. The German one has more than 1,6 million articles and more than 4,5 million redirect pages.

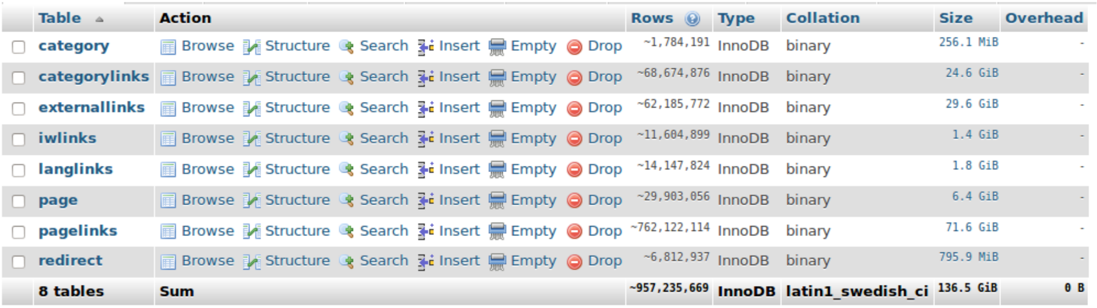


	Table	Action						Rows	Type	Collation	Size	Overhead
□	**category**	Browse	Structure	Search	Insert	Empty	Drop	~1,784,191	InnoDB	binary	256.1 MiB	-
□	**categorylinks**	Browse	Structure	Search	Insert	Empty	Drop	~68,674,876	InnoDB	binary	24.6 GiB	-
□	**externallinks**	Browse	Structure	Search	Insert	Empty	Drop	~62,185,772	InnoDB	binary	29.6 GiB	-
□	**iwlinks**	Browse	Structure	Search	Insert	Empty	Drop	~11,604,899	InnoDB	binary	1.4 GiB	-
□	**langlinks**	Browse	Structure	Search	Insert	Empty	Drop	~14,147,824	InnoDB	binary	1.8 GiB	-
□	**page**	Browse	Structure	Search	Insert	Empty	Drop	~29,903,056	InnoDB	binary	6.4 GiB	-
□	**pagelinks**	Browse	Structure	Search	Insert	Empty	Drop	~762,122,114	InnoDB	binary	71.6 GiB	-
□	**redirect**	Browse	Structure	Search	Insert	Empty	Drop	~6,812,937	InnoDB	binary	795.9 MiB	-
	8 tables	**Sum**						**~957,235,669**	**InnoDB**	**latin1_swedish_ci**	**136.5 GiB**	**0 B**

Figure 6.1: The English Wikipedia database

6.2 Data Sets

Two data sets are used to evaluate the performance of the proposed approach for document clustering.

1. **Reuters 21578.**[1] One of the most widely used collection of documents for text clustering and categorization research. The documents from this collection appeared on the Reuters newswire in 1987. The documents were assembled and indexed with categories manually by personnel from Reuters Ltd. and Carnegie Group, Inc. For the purpose of this Master's Thesis the non-labeled documents are removed and only these with unique categories are extracted. A set of 2571 document distributed in 59 categories is used.

2. **Collection of news in German and English as a comparable corpus**. A comparable corpus is a set of similar documents written in more than one language or variety. Therefore, different news is obtained from various web pages, which offer documents in German and English simultaneously. Press releases were gathered from the BMW Group page[2], Merck Group[3] and Schwarzkopf[4]. The documents are equally distributed and represent texts with different styles and areas. The BMW and Merck documents are mostly news from the technology world, while these from Schwarzkopf are from the area of cosmetics and beauty.

6.3 Evaluation Metric

For the next experiments three evaluation functions are used: *averageInClusterSum*, *averageOutClusterSum* and *ratio* between both of them.

The *averageInClusterSum* (cf. Figure: 6.2) represents the average sum of the similarities (defined in Section 4.3) of all documents in a cluster. In the best case, the sum converges to 1, meaning that the documents in one cluster are similar in respect to content and meaning.

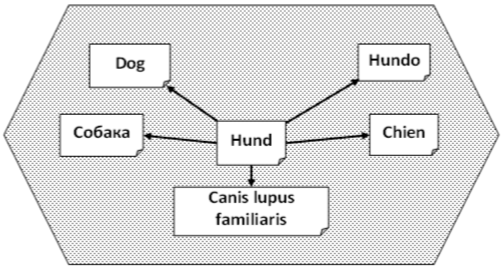


Figure 6.2: Average In Cluster Sum

[1] http://www.daviddlewis.com/resources/testcollections/reuters21578/

[2] https://www.press.bmwgroup.com/pressclub/p/pcgl/startpage.html

[3] http://www.merckgroup.com/en/media/news_releases/news_releases.html

[4] http://www.schwarzkopf.com/sk/en/home.html

The *averageOutClusterSum* (cf. Figure: 6.3) is the average sum of the similarities (defined in Section 4.3) of a document in a cluster with all other documents outside of this cluster. Given best scenario, the sum converges to 0, meaning that the documents from different clusters are as distinct as possible.

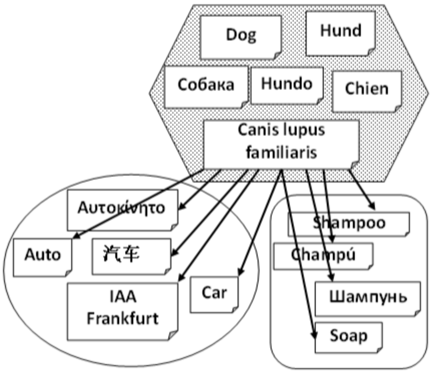


Figure 6.3: Average Out Cluster Sum

The *ratio* between both functions ($averageInClusterSum/averageOutClusterSum$) is considered as an important measure for choosing the better similarity functions. It gives a sense of the efficiency of these functions. The larger the *ratio* value, the better the results. For instance, to compare two similarity functions, the *averageInClusterSum* and *averageOutClusterSum* are estimated for both of them as well as their *ratios,* respectively. Then, if the first function delivers for almost all documents similarity values equal to 1 and both its *averageInClusterSum* and *averageOutClusterSum* converge to 1, then the *ratio* will be very close to 1. On the contrary, if the second function's *averageInClusterSum* is equal to $0,91$ and its *averageOutClusterSum* has a value of $0,003$, then the *ratio* will be equal to $303,33$. The comparison of the ratios helps to consider which function delivers better similarity results. In this example, the second function is better, although its *averageInClusterSum* has smaller values.

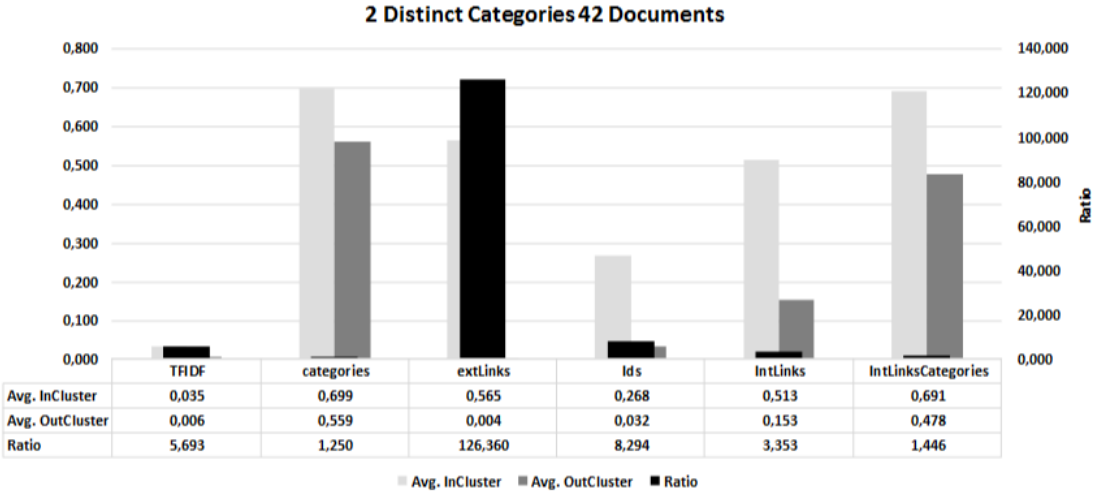
In the next sections results from experiments with the similarity functions will be presented.

6.4 Similarity Functions in One Language

First, the similarity functions for documents written in one language (English) are tested. Five experiments are conducted with different data sets from the Reuters database. These evaluations show how the distance functions change when the topics are similar or different and when the number of documents is fairly small or large.

6.4.1 Experiment №1

In this experiment a subset of the Reuters documents containing the topics *coffee* and *gold* is taken. There are 42 documents in both of the topics, distributed equally. This test reveals the results of the similarity functions when the number of documents is small and the topics are distant. (cf. Figure: 6.4)



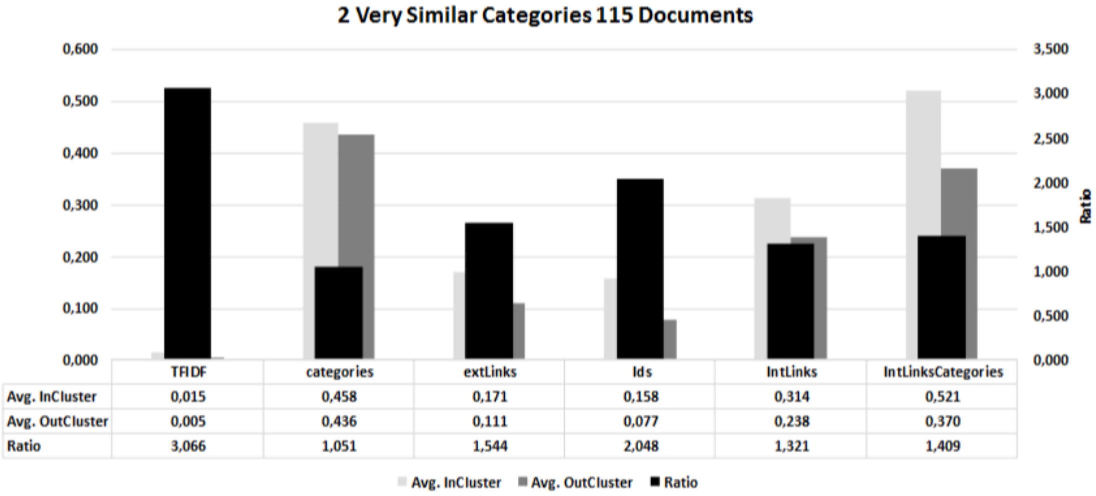
	TFIDF	categories	extLinks	Ids	IntLinks	IntLinksCategories
Avg. InCluster	0,035	0,699	0,565	0,268	0,513	0,691
Avg. OutCluster	0,006	0,559	0,004	0,032	0,153	0,478
Ratio	5,693	1,250	126,360	8,294	3,353	1,446

Figure 6.4: Evaluation results of coffee and gold categories.

From the performance results in Figure 6.4, it can be seen that within all similarity functions the best results are delivered using the external links and the Id of Wikipedia.

6.4.2 Experiment №2

In this experiment a subset containing 115 documents from the topics *money-fx* and *money-supply* is taken. It shows how the functions act on small number of documents from very similar categories.



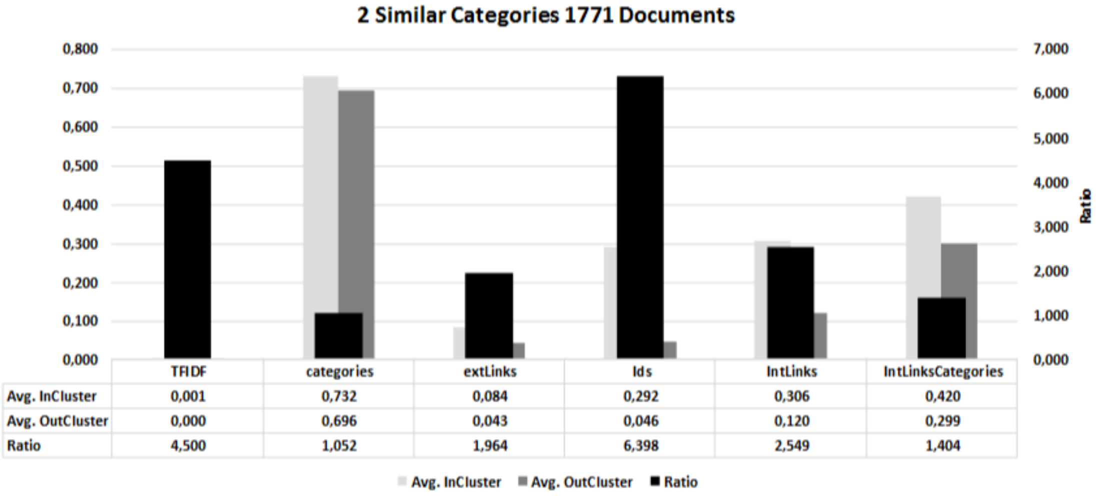
	TFIDF	categories	extLinks	Ids	IntLinks	IntLinksCategories
Avg. InCluster	0,015	0,458	0,171	0,158	0,314	0,521
Avg. OutCluster	0,005	0,436	0,111	0,077	0,238	0,370
Ratio	3,066	1,051	1,544	2,048	1,321	1,409

Figure 6.5: Evaluation results of money-fx and money-supply categories.

From Figure 6.5, it can be observed that more precise results are received from the TFIDF, the Ids and external links of Wikipedia.

6.4.3 Experiment №3

Here the experiment provides results from two similar categories with large number of documents. The categories are *earn* and *acq* and there are 1771 documents in this collection.



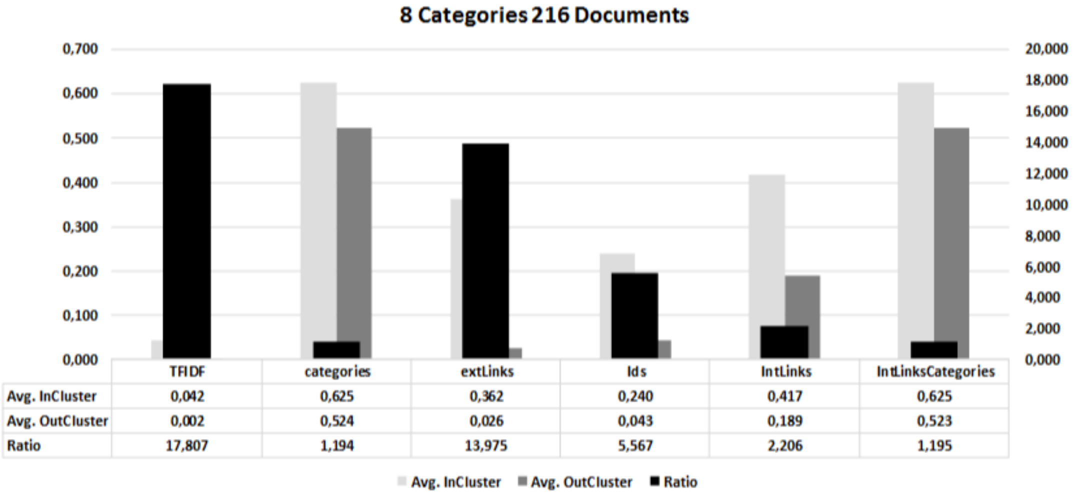
	TFIDF	categories	extLinks	Ids	IntLinks	IntLinksCategories
Avg. InCluster	0,001	0,732	0,084	0,292	0,306	0,420
Avg. OutCluster	0,000	0,696	0,043	0,046	0,120	0,299
Ratio	4,500	1,052	1,964	6,398	2,549	1,404

Figure 6.6: Evaluation results of earn and acq categories.

Figure 6.6 demonstrates again that the Ids, TFIFD and the external links of Wikipedia are good in the document clustering in one language.

6.4.4 Experiment №4

For this experiment 8 categories with equally distributed 216 documents are picked. (cf. Figure: 6.7)



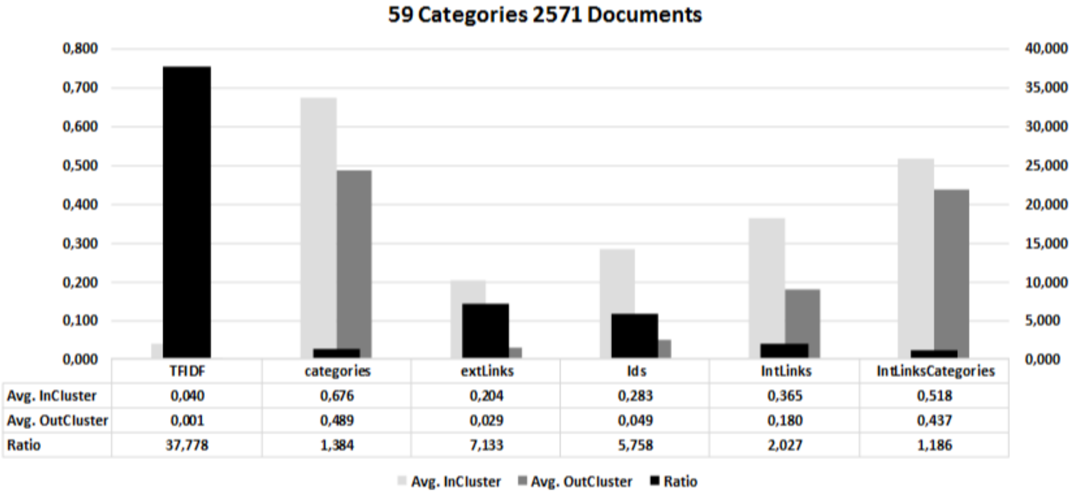
	TFIDF	categories	extLinks	Ids	IntLinks	IntLinksCategories
Avg. InCluster	0,042	0,625	0,362	0,240	0,417	0,625
Avg. OutCluster	0,002	0,524	0,026	0,043	0,189	0,523
Ratio	17,807	1,194	13,975	5,567	2,206	1,195

Figure 6.7: Evaluation results of coffee, gold, trade, grain, ship, sugar, alum and gas categories.

It can be seen again that the TFIFD, the external links and Ids of Wikipedia provide significantly better results than the other three functions.

6.4.5 Experiment №5

Now all 2571 documents within 59 categories are used.



	TFIDF	categories	extLinks	Ids	IntLinks	IntLinksCategories
Avg. InCluster	0,040	0,676	0,204	0,283	0,365	0,518
Avg. OutCluster	0,001	0,489	0,029	0,049	0,180	0,437
Ratio	37,778	1,384	7,133	5,758	2,027	1,186

Figure 6.8: Evaluation results of all documents.

Figure 6.8 shows once more that the TFIDF, external links and Ids of Wikipedia deliver the best results. It is also visible that when the number of documents is large and they are coming from very different topics, using various language styles and divergent spectrum of terms, the TFIDF method works better.

6.5 Multilingual Similarity Functions

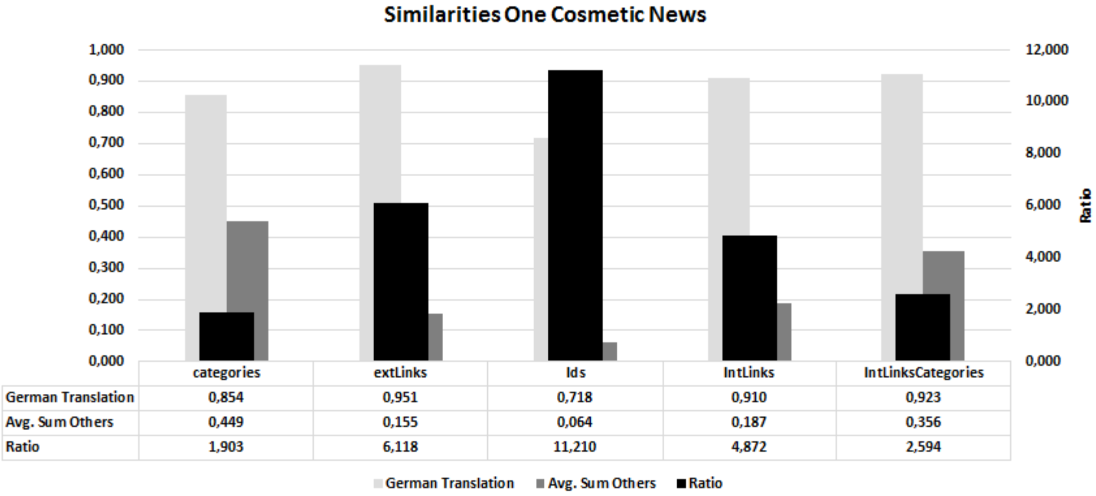
Four experiments are conducted on the comparable corpus. In the first experiment, a document, written in English, is taken from the Schwarzkopf data set and its similarities with all news in German are calculated. For the second test a document, written in English is picked from BMW AG news data set and the same procedure is executed. In the third experiment the average sums of all translation is compared and in the fourth the results of the *averageInClusterSum* and the *averageOutClusterSum* are observed.

6.5.1 Experiment №1

For this experiment, an English document related to the cosmetic industry is taken. The German translation of its content is found and extracted from Wikipedia and its enrichment vectors are populated with data from the German version of Wikipedia. Afterwards, the similarities with all other German documents are calculated. Table 6.1 and Figure 6.9 depict the results. The best similarity scores (in bold) are received from the external links and the Ids of Wikipedia, with the German translation of the news. It can also be seen, that the cosmetic article has similarity values nearly equal to 0 compared to the news provided from BMW and Merck. On the contrary, the similarities of this article with the rest news from Schwarzkopf are higher. As a result, when a cluster is built, it will contain news mainly from Schwarzkopf.

Source	Categories	extLinks	Ids	IntLinks	IntLinksCategories
BMW AG	0,446	0,003	0,001	0,059	0,289
BMW AG	0,394	0	0	0,056	0,197
BMW AG	0,418	0	0	0,076	0,272
BMW AG	0,377	0	0,002	0,068	0,260
BMW AG	0,415	0	0	0,082	0,269
Schwarzkopf	0,498	0,462	0,182	0,379	0,444
Schwarzkopf	**0,854**	**0,951**	**0,718**	**0,910**	**0,923**
Schwarzkopf	0,621	0,811	0,357	0,730	0,770
Schwarzkopf	0,576	0,670	0,271	0,527	0,530
Schwarzkopf	0,440	0,149	0,054	0,207	0,421
Merck KGaA	0,406	0	0,023	0,061	0,218
Merck KGaA	0,394	0	0	0,050	0,238
Merck KGaA	0,448	0,036	0	0,105	0,415
Merck KGaA	0,426	0,025	0,005	0,115	0,331
Merck KGaA	0,426	0,018	0,001	0,1	0,326

Table 6.1: Results of all similarity functions for a cosmetic news.



	categories	extLinks	Ids	IntLinks	IntLinksCategories
German Translation	0,854	0,951	0,718	0,910	0,923
Avg. Sum Others	0,449	0,155	0,064	0,187	0,356
Ratio	1,903	6,118	11,210	4,872	2,594

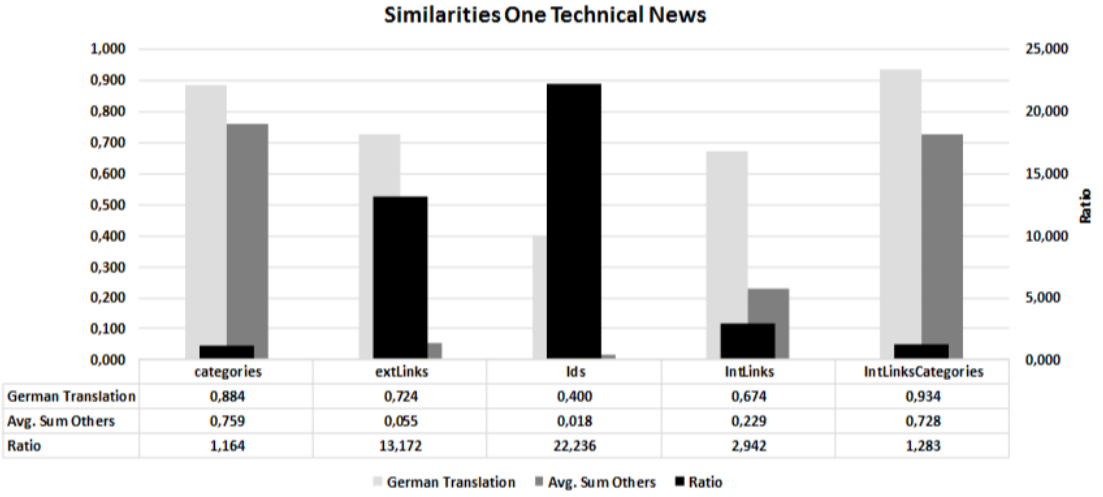
Figure 6.9: Comparison between the similarity functions of the German translation and all other German documents.

6.5.2 Experiment №2

Here one English article containing mostly technical information and terms is taken from the BMW AG and compared with all German documents. From Table 6.2 and Figure 6.10 it can be easily observed that the best result are delivered again from the external links and Ids. The row with the bold values is the German translation of the English document. The similarity scores for all articles containing information from the area of cosmetics and beauty are approximately 0.

Source	Categories	extLinks	Ids	IntLinks	IntLinksCategories
BMW AG	0,859	0,132	0,013	0,203	0,792
BMW AG	0,759	0,043	0,045	0,239	0,629
BMW AG	**0,884**	**0,724**	**0,400**	**0,674**	**0,934**
BMW AG	0,727	0,029	0,019	0,308	0,837
BMW AG	0,803	0,099	0,059	0,353	0,833
Schwarzkopf	0,822	0	0,016	0,200	0,669
Schwarzkopf	0,587	0,001	0	0,119	0,443
Schwarzkopf	0,657	0	0	0,137	0,524
Schwarzkopf	0,720	0	0,004	0,143	0,753
Schwarzkopf	0,726	0,017	0,007	0,167	0,743
Merck KGaA	0,770	0,278	0,044	0,324	0,758
Merck KGaA	0,741	0,004	0	0,205	0,781
Merck KGaA	0,846	0,002	0,003	0,214	0,788
Merck KGaA	0,794	0,122	0,024	0,316	0,803
Merck KGaA	0,814	0,042	0,015	0,280	0,834

Table 6.2: Results of all similarity functions for a cosmetic news.

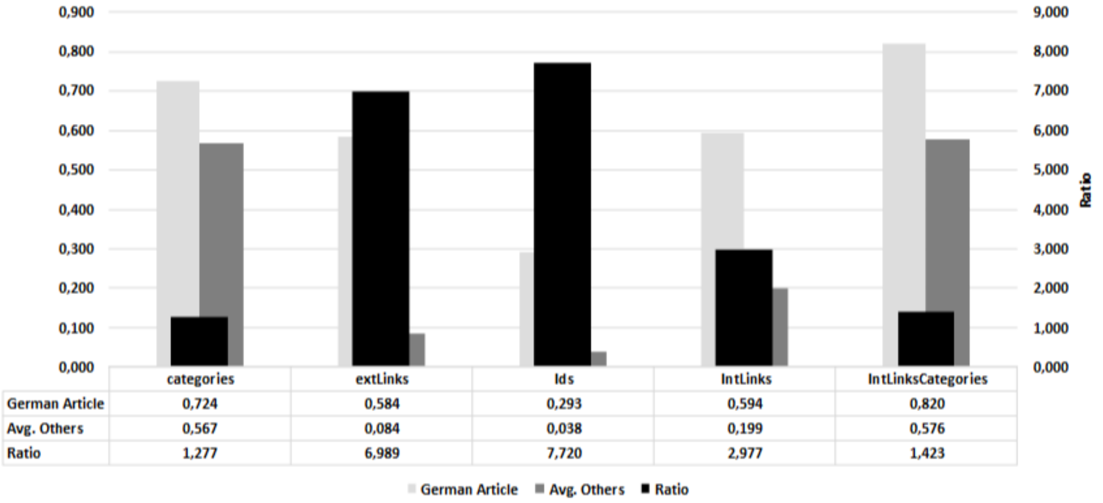


	categories	extLinks	Ids	IntLinks	IntLinksCategories
German Translation	0,884	0,724	0,400	0,674	0,934
Avg. Sum Others	0,759	0,055	0,018	0,229	0,728
Ratio	1,164	13,172	22,236	2,942	1,283

Figure 6.10: Comparison between the similarities of the German translation to a technical news and all other German documents.

6.5.3 Experiment №3

In this experiment the similarities among all English and German news are computed and the average sums are depicted in Figure 6.11. It is again noticeable that the external links and Ids deliver the best results. In all cases, the documents with the highest similarity values are the German translation of the English documents.

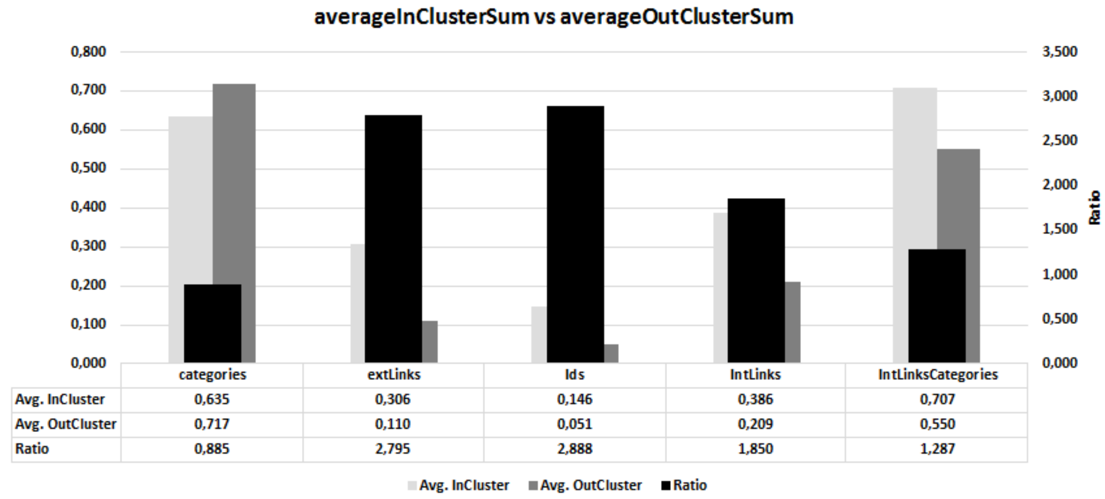


	categories	extLinks	Ids	IntLinks	IntLinksCategories
German Article	0,724	0,584	0,293	0,594	0,820
Avg. Others	0,567	0,084	0,038	0,199	0,576
Ratio	1,277	6,989	7,720	2,977	1,423

Figure 6.11: Comparison between the average similarities of the German translations and all other German articles.

6.5.4 Experiment №4

Now the inCluster and outCluster sums of all news are compared. Figure 6.12 proves again that the external links and the Ids are the most reliable similarity measures for multilingual clustering.



	categories	extLinks	Ids	IntLinks	IntLinksCategories
Avg. InCluster	0,635	0,306	0,146	0,386	0,707
Avg. OutCluster	0,717	0,110	0,051	0,209	0,550
Ratio	0,885	2,795	2,888	1,850	1,287

Figure 6.12: The averageInClusterSum and averageOutClusterSum of the similarities for all documents.

Chapter 7

Parameters Setting

As previously stated in Chapter 4 the weight parameters from equation 4.1 must be optimized in order to get the best results. From the conducted experiments in the previous chapter, it is noticeable that for different data sets, the similarity functions deliver different results.

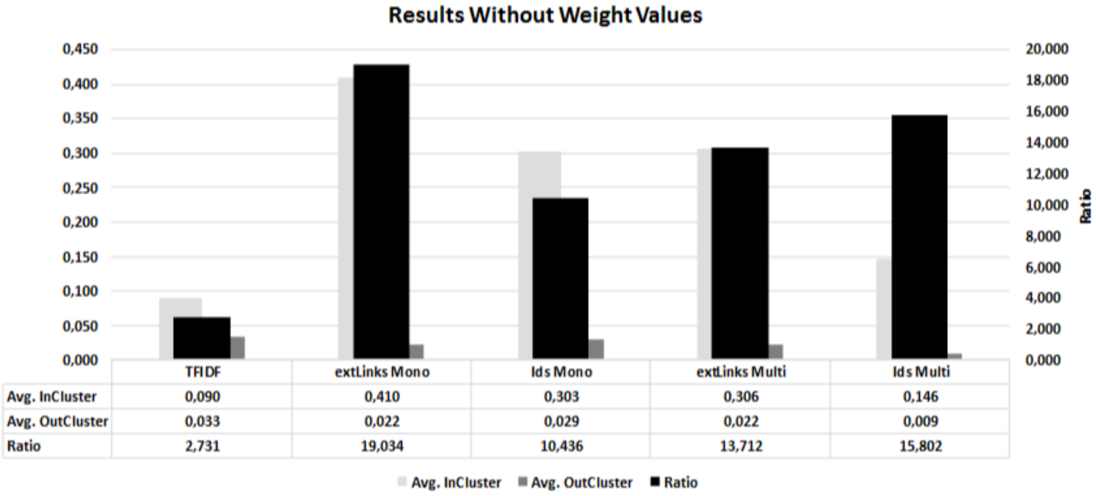
It can be easily concluded that in all experiments, the functions based on the categories, internal links and the internal links' categories of Wikipedia do not provide significant improvement to the TFIDF measure in the single language case and are worse than the rest Wikipedia enrichments in all multilingual experiments. Therefore, their weights are set to 0. As a result, the equation 4.1 is simplified to:

$$sim(d_n,d_m) = \alpha \cdot sim(d_n,d_m)^{TFIDF} + \beta \cdot sim(d_n,d_m)^{extLinks} + \gamma \cdot sim(d_n,d_m)^{Ids}. \qquad (7.1)$$

Where $\alpha+\beta+\gamma=1$. Based on the results from the monolingual experiments in Chapter 6, it can be deduced that when clustering between documents from similar topics is done, $sim(d_n,d_m)^{TFIDF}$ and $sim(d_n,d_m)^{Ids}$ outperform the other similarity functions. If texts from distinct topics are clustered, then the function $sim(d_n,d_m)^{extLinks}$ provides better results.

When the clustering is done in multilingual collection of documents, computing the TFIDF similarity does not provide correct results, as most of the words differ notably. Consequently, only the Wikipedia similarity measures are used and as seen in the experiment, best results are obtained from the function $sim(d_n,d_m)^{Ids}$, followed by $sim(d_n,d_m)^{extLinks}$, which to some extent has lower *ratio* than $sim(d_n,d_m)^{Ids}$.

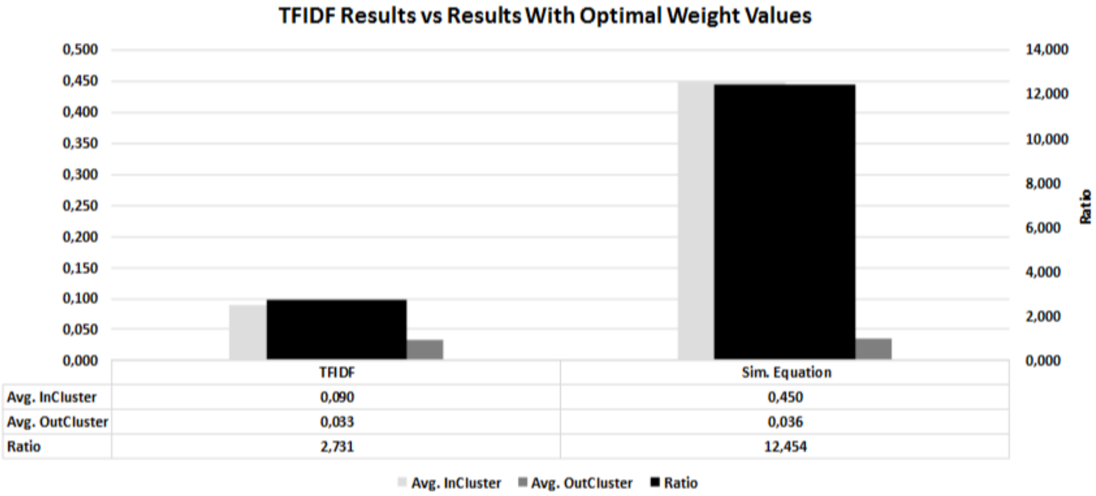
Another experiment is executed. The collection of multilingual news from Chapter 6 is used as a data set. Here, the purpose is to perform the clustering simultaneously on both languages. Firstly, the similarities among all documents in the base language are calculated (in this test the English articles). Then the similarities among all texts, written in the base language, with the other documents, written in other languages, are found. Finally all results are taken, compared together and the clusters are formed. Based on the results (cf. Figure: 7.1), tuning on equation: 8 is performed in order to find the best weight values. This is done by increasing each weight value from $0,1$ to 1 with a step equal to $0,1$.



	TFIDF	extLinks Mono	Ids Mono	extLinks Multi	Ids Multi
Avg. InCluster	0,090	0,410	0,303	0,306	0,146
Avg. OutCluster	0,033	0,022	0,029	0,022	0,009
Ratio	2,731	19,034	10,436	13,712	15,802

Figure 7.1: Results from all similarity functions.

Two different weights' settings are used. When the similarity between monolingual documents is estimated, then best results are achieved with weight values $\alpha = 0,6$, $\beta = 0,2$ and $\gamma = 0,2$. When two multilingual articles are compared, the weight of the TFIDF function is set to 0, e.g. $\alpha = 0$, and optimal results are obtained with weights $\beta = 0,4$ and $\gamma = 0,6$. Figure 7.2 depicts the comparison between the *averageInClusterSum* and the *averageOutClusterSum* values of the TFIDF function, clustering all English documents, and equation 7.1 with the optimal weights, clustering the English and the German text simultaneously. It is notable that the similarity equation delivers significant improvement over the TFIDF function, although it clusters twice as many documents and in different languages.



	TFIDF	Sim. Equation
Avg. InCluster	0,090	0,450
Avg. OutCluster	0,033	0,036
Ratio	2,731	12,454

Figure 7.2: Comparison between the TFIDF results and the results from equation 7.1 with optimal weight values.

Chapter 8

Conclusion and Future Work

In this Master's Thesis, a method for enhancing the performance of multilingual document clustering is presented. Its main advantage is that it does not depend on multilingual resources like dictionaries, machine translation systems or thesaurus. It uses Wikipedia as an encyclopedic knowledge base to improve the document's representation. Wikipedia is by far the largest encyclopedia in existence, but unfortunately, is not structured as some other thesaurus. Therefore, an approach to extract synonyms, hypernyms and associative relations (ontology information) for the content of a document through analyzing the rich links in Wikipedia is demonstrated. Consequently, five different enrichments are retrieved and tested. The external links and Ids of the articles from Wikipedia have proved to be very useful, not only in the translation, but also in forming clusters of documents, written in the same language. Incorporating this background knowledge into the document representation overcomes the limitation of the "bag of words" method. This approach is very extensible since it is easy to use the most up to date version of Wikipedia, making it possible to handle the newest events and terms, as well as to integrate new languages by just importing their database dumps.

However, there is a room for improvement. From the experiments, it is noticeable that some similarity functions deliver better results when the collection of documents contains large number of articles and/or many unique terms. With further research, a higher performance can be achieved by introducing another weight to the similarity equation indicating the size of the collection. Moreover, different strategy for clustering multilingual documents can be developed. First clusters can be formed for each language from the collection. Then, through the crosslingual vector similar documents can be matched and their clusters combined. While it is pretty straightforward to adapt this new strategy for multilingual text clustering problems, there are still open questions as to whether it can combine the right clusters or will bring additional noise.

List of Figures

List of Tables

Appendix A

My.cnf Configuration

```
# The following options will be passed to all MySQL clients
[client]
#password        = your_password
port             = 3306
socket           = /opt/lampp/var/mysql/mysql.sock

# The MySQL server
[mysqld]
user             = nobody
port             = 3306
socket           = /opt/lampp/var/mysql/mysql.sock

skip-external-locking

key_buffer = 1100M
max_allowed_packet = 1600M
table_cache = 1024
sort_buffer_size = 1024M
net_buffer_length = 64K
read_buffer_size = 1800M
read_rnd_buffer_size = 1512M
myisam_sort_buffer_size = 64M

# Where do all the plugins live
plugin_dir = /opt/lampp/lib/mysql/plugin/
# required unique id between 1 and 2^32 - 1
# defaults to 1 if master-host is not set
# but will not function as a master if omitted
server-id        = 1
innodb_data_home_dir = /opt/lampp/var/mysql/
innodb_data_file_path = ibdata1:10M:autoextend
innodb_log_group_home_dir = /opt/lampp/var/mysql/
```

```
innodb_buffer_pool_size = 1536M
innodb_additional_mem_pool_size = 1812M
innodb_log_file_size = 1600M
innodb_log_buffer_size = 130M
innodb_flush_log_at_trx_commit = 2
innodb_lock_wait_timeout = 50


[mysqldump]
quick
max_allowed_packet = 512M


[mysql]
no-auto-rehash


[isamchk]
key_buffer = 1024M
sort_buffer_size = 1024M
read_buffer = 1024M
write_buffer = 1024M

[myisamchk]
key_buffer = 1024M
sort_buffer_size = 1024M
read_buffer = 1024M
write_buffer = 1024M

[mysqlhotcopy]
interactive-timeout
```

Bibliography

[1] SF Adafre and M De Rijke. Finding similar sentences across multiple languages in wikipedia. 2006.

[2] Eric Brill and RJ Mooney. An overview of empirical natural language processing. *AI magazine*, 18(4):13–24, 1997.

[3] JF da Silva and GP Lopes. A Statistical Approach for Multilingual Document Clustering and Topic Extraction from Clusters. 2004.

[4] DK Evans, JL Klavans, and KR McKeown. Columbia newsblaster: multilingual news summarization on the Web. 2004.

[5] Evgeniy Gabrilovich and Shaul Markovitch. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. Technical report, 2006.

[6] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. Technical report, 2007.

[7] Jim Giles. Internet encyclopaedias go head to head. *Nature*, 438(7070):900–1, December 2005.

[8] Andreas Hotho. Wordnet improves Text Document Clustering. 2003.

[9] Andreas Hotho, Alexander Maedche, and Steffen Staab. Text Clustering Based on Good Aggregations University of Karlsruhe University of Karlsruhe. pages 1–2, 2001.

[10] Jian Hu, Lujun Fang, Yang Cao, Hua-Jun Zeng, Hua Li, Qiang Yang, and Zheng Chen. Enhancing text clustering by leveraging Wikipedia semantics. Technical report, New York, New York, USA, 2008.

[11] Xiaohua Hu, Xiaodan Zhang, Caimei Lu, E. K. Park, and Xiaohua Zhou. Exploiting Wikipedia as external knowledge for document clustering. Technical report, New York, New York, USA, 2009.

[12] Michael Kende. How the Internet continues to sustain growth and innovation October 2012. (October), 2012.

[13] K Kumar, GSK Santosh, and Vasudeva Varma. Multilingual document clustering using wikipedia as external knowledge. *Multidisciplinary Information Retrieval*, 2011.

[14] Y Li, WPR Luk, KSE Ho, and FLK Chung. Improving weak ad-hoc queries using wikipedia asexternal corpus. 2007.

[15] Mary Meeker, S Devitt, and L Wu. Internet trends. 2010.

[16] Gerard De Melo and Gerhard Weikum. Untangling the cross-lingual link structure of Wikipedia. (July), 2010.

[17] David Milne. Computing semantic relatedness using wikipedia link structure. Technical report, 2007.

[18] D Nguyen and A Overwijk. WikiTranslate: query translation for cross-lingual information retrieval using only Wikipedia. 2009.

[19] Martin Potthast, Benno Stein, and Maik Anderka. A Wikipedia-based multilingual retrieval model. *Advances in Information Retrieval*, pages 522–530, 2008.

[20] Bruno Pouliquen, Ralf Steinberger, and Camelia Ignat. Multilingual and cross-lingual news topic tracking. 2004.

[21] Juan Ramos. Using tf-idf to determine word relevance in document queries. 2003.

[22] Stephen Robertson. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation*, 60(5):503–520, 2004.

[23] J Voss. Measuring wikipedia. pages 1–12, 2005.

[24] Pu Wang and Carlotta Domeniconi. Building semantic kernels for text classification using wikipedia. Technical report, 2008.

[25] Pu Wang, Jian Hu, Hua-Jun Zeng, Lijun Chen, and Zheng Chen. Improving Text Classification by Using Encyclopedia Knowledge. Technical report, October 2007.

[26] Chih-Ping Wei, Christopher C. Yang, and Chia-Min Lin. A Latent Semantic Indexing-based approach to multilingual document clustering. *Decision Support Systems*, 45(3):606–620, June 2008.

[27] Dani Yogatama and K Tanaka-Ishii. Multilingual spectral clustering using document similarity propagation. (August), 2009.

[28] Torsten Zesch and Iryna Gurevych. Analysis of the Wikipedia category graph for NLP applications. 2007.

[29] Torsten Zesch, Christof Müller, and Iryna Gurevych. Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. *LREC*, 2008.

[30] Y Zhao and G Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, pages 311–331, 2004.