
Entwicklung eines semantischen Browsers für Linked Open Data

Developing a Semantic Browser for Linked Open Data
Bachelor-Thesis von Alexander Seeliger aus Bad Hersfeld
Oktober 2012



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Informatik
Knowledge Engineering

Entwicklung eines semantischen Browsers für Linked Open Data
Developing a Semantic Browser for Linked Open Data

Vorgelegte Bachelor-Thesis von Alexander Seeliger aus Bad Hersfeld

1. Gutachten: Prof. Johannes Fürnkranz
2. Gutachten:

Tag der Einreichung:

Erklärung zur Bachelor-Thesis

Hiermit versichere ich, die vorliegende Bachelor-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 30. Oktober 2012

(Alexander Seeliger)

Inhaltsverzeichnis

1	Einführung und Grundlagen	3
1.1	Einführung und Motivation	3
1.2	Linked Open Data	3
1.3	Resource Description Framework	4
1.4	DBpedia	5
2	Existierende semantische Browser für Linked Open Data	7
2.1	DBpedia Webzugriff	7
2.2	Disco - Hyperdata Browser	7
2.3	Tabulator	8
2.4	aemoo	8
2.5	Freebase Webzugriff	9
2.6	Zusammenfassung	10
3	Entwicklung eines semantischen Browsers	11
3.1	Konzept der automatisierten Gruppierung	11
3.2	Datensätze in RDF	11
3.3	Semantische Äquivalenz von Wörtern	13
3.4	Clustering	15
3.5	Gruppentitel	17
3.6	Implementierung	17
3.7	Optimierungen	19
4	Evaluierung	21
4.1	Testinstanzen und -gruppen	21
4.2	Ablauf der Benutzerstudie	23
4.3	Auswertung	23
4.4	Feedback von Testern	26
4.5	Performance	27
5	Ausblick	29
6	Fazit	31
7	Anhang	36

1 Einführung und Grundlagen

1.1 Einführung und Motivation

Das Internet ist das größte Informationsmedium auf der Welt und umfasst eine fast unendliche Anzahl an Informationen. Die Aktualität und Geschwindigkeit, in der Informationen bereitgestellt werden können, lässt sich derzeit von keinem anderen Medium als dem Internet übertreffen. Informationen lassen sich innerhalb weniger Augenblicke auf den heimischen Rechner oder unterwegs auf das Smartphone laden. Das Internet ist aus unserem Alltag nicht mehr wegzudenken. Doch die Art und Weise wie Daten im Internet bereitgestellt werden, macht es zu einem sehr unüberschaubarem Medium. Ein Ansatz, der diese Menge an Daten ordnen und den Zugriff darauf vereinfachen will, ist das *Semantic Web*. Bislang war es nur mit großem Aufwand möglich, mittels einer Suchmaschine eine direkte Antwort auf eine Frage zu erhalten. Anstatt die Antwort direkt zu kennen, liefert eine Suchmaschine nur eine Liste von Webseiten, die die eingegebenen Wörter enthalten. Dieser Ansatz stößt mittlerweile an seine Grenzen, weshalb das Semantic Web entwickelt worden ist. Idee hierbei ist, Informationen bereits bei der Erzeugung in einer für den Computer verständliche Art und Weise zu speichern. Somit kann die Suche nach Informationen in Zukunft effizienter und zielführender ermöglicht werden.

Es existieren bereits eine Reihe von Anwendungen, die semantische Daten dazu verwenden, dem Benutzer konkrete Informationen zu liefern. 2009 wurde *Wolfram Alpha* ins Leben gerufen, welches durch Eingabe von Fragen in natürlicher Sprache Antworten zu diversen Themenbereichen liefert [Wol, 2012]. Eine ähnliche semantische Datenbank (*Knowledge Graph*) wird derzeit von Google aufgebaut. Der Knowledge Graph liefert neben den Suchergebnissen konkrete Informationen zu Personen in Form einer kurzen Zusammenfassung [Singhal, 2012]. Die Funktionsweise beider Ansätze ist nicht genau bekannt.

Ein offenerer Ansatz ist das dezentrale *Linked Open Data* [Berners-Lee, 2006]. Es ist ein semantisches Netz, welches Informationen (Fakten) zu verschiedenen Themenbereichen (unter anderem zu Medien, Geographie, Publikationen, Regierungen und Medizin) enthält. Linked Open Data wird dabei den Grundprinzipien des Internets - der uneingeschränkte Zugriff und die Möglichkeit eigene Informationen bereitzustellen - gerecht. Es ermöglicht die Speicherung beliebiger Informationen und kennt zum Beispiel das Geburtsdatum von berühmten Personen, die Schauspieler eines bestimmten Films oder die Einwohnerzahl einer Stadt. Anstatt die Informationen für den Menschen als Fließtext zu speichern, werden die Daten so abgelegt, dass der Computer sie verstehen und verarbeiten kann. Linked Open Data verwendet Technologien des Semantic Web und ist demnach ein großer Teil dieses.

Da die Informationen im Linked Open Data hauptsächlich für den Computer speziell kodiert werden, muss eine Möglichkeit geschaffen werden, diese dem Nutzer sinnvoll und verständlich anzuzeigen. Eine einheitliche und benutzerfreundliche Anzeige der Fakten im Linked Open Data wurde bislang noch nicht umgesetzt. In dieser Arbeit sollen daher Ansätze untersucht werden, wie beliebige Fakten aus dem Linked Open Data nach dem Sinnzusammenhang gruppiert werden können. Dazu sollen verschiedene Möglichkeiten gefunden werden, wie die Semantik der Daten erkannt werden kann und welche Cluster-Algorithmus geeignet sind, um diese Datenmenge zu gruppieren. Schließlich soll die beste Lösung in einen bereits existierenden Browser für Linked Open Data integriert werden.

1.2 Linked Open Data

Linked Open Data ist, wie bereits erwähnt, ein offenes dezentrales semantisches Netz. Dessen Ziel ist es, dass jeder Daten konsumieren, sowie produzieren kann. Tim Berners-Lee hat Linked Open Data mit seinen 4 Grundprinzipien [Berners-Lee, 2006] stark geprägt:

1. „Use URIs as names for things“

-
2. „Use HTTP URIs so that people can look up those names.“
 3. „When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)“
 4. „Include links to other URIs. so that they can discover more things.“

Die Offenheit und die starke Vernetzung zwischen verschiedenen Datenquellen, machen Linked Open Data zu einer sehr mächtigen Informationsquelle. Im Gegensatz zu Wolfram Alpha oder Knowledge Graph kann jeder hier beliebig viele Informationen bereitstellen und diese zu bereits vorhandenen Datensätzen verlinken. Durch die Dezentralisierung und die einfache Bereitstellung neuer Datensätze, wächst das Netz ständig mit diesen neuen Informationen.

Der Zugriff auf Linked Open Data ist offen und jeder mit einem Webbrowser kann eine beliebige Linked Open Data Quelle durch Eingabe einer URL aufrufen. Die meisten Quellen besitzen eine HTML-Ansicht, sodass ein Blick auf die Datensätze möglich ist. Die Webansichten sind in der Regel sehr rudimentär gehalten und sprechen in erster Linie Entwickler an. Einige dieser Ansichten werden im Abschnitt 2 vorgestellt.

Im Linked Open Data setzt man auf bewährte Technologien des Semantic Web, wovon die meisten davon standardisiert sind. Zur Speicherung der Daten wird das *Resource Description Framework* (kurz RDF) verwendet. Damit lassen sich auch komplexe Datenstrukturen in einem einheitlichen Datenformat bereitstellen.

1.3 Resource Description Framework

RDF ist ein Standard des World Wide Web Consortiums (W3C) [Klyne and Carroll, 2004] und ist recht weit verbreitet. Ein RDF-Dokument beschreibt einen gerichteten Graphen, bei welchem Knoten und Kanten eindeutige Namen besitzen [Hitzler et al., 2007]. Mit Graphen lassen sich im Gegensatz zu der Beschreibungssprache XML auch nicht hierarchische Informationen beschreiben. RDF wurde für das Web entwickelt, um allgemeine Beziehungen zwischen Ressourcen zu beschreiben, ohne dass eine hierarchische Struktur notwendig ist. Die Graphenstruktur erlaubt es darüber hinaus, die Speicherung von Informationen dezentral zu ermöglichen. Mehrere Graphen lassen sich zusammenführen und bilden demnach eine größere Menge an Informationen, die sich aber auch effizient getrennt verarbeiten lassen [Hitzler et al., 2007, S. 36].

Damit mit RDF eine Ressource innerhalb einer Komposition von Graphen eindeutig identifiziert werden kann, verwendet man im RDF ein *Uniform Resource Identifier* (kurz URI [Berners-Lee, 2005b]). Man vergibt jedem klar identifizierbaren Objekt ein URI, um es als Ressource beschreiben zu können. Die URI dienen demnach als Referenz auf die tatsächlich gemeinten Dinge [Hitzler et al., 2007, S. 28]. Neben Ressourcen gibt es noch die Möglichkeit primitive Datenwerte in RDF abzulegen. Diese Werte nennt man *Literale* und ermöglichen die Speicherung diverser Datentypen¹. Literale sind stets mit einer Ressource durch eine Kante verbunden.

Die von RDF beschriebenen gerichteten Graphen stellt man üblicherweise als Menge von vorhandenen Kanten als Tripel dar. Ein Tripel besteht aus Subjekt, Prädikat und Objekt. Bei Subjekten handelt es sich immer um eine konkrete Ressource, zu denen eine Information gespeichert werden soll. Das Prädikat beschreibt, um was es sich für eine Information handelt und das Objekt ist entweder ein Literal oder eine andere Ressource.

Eine weit verbreitete und kompakte Notation zur besseren Lesbarkeit von RDF ist die *Notation 3* (N3) [Berners-Lee, 2005a], die auch in dieser Arbeit verwendet wird. Sie ist kein Standard von RDF wird aber von vielen Tools unterstützt. Die N3 besteht aus den 3 Werten in der Reihenfolge Subjekt, Prädikat

¹ Unterstützt werden fast alle Datentypen von XML-Schema (Abschnitt 3.3 [Graham Klyne, 2004])

und Objekt gefolgt von einem Punkt. Mit einem Komma lassen sich mehrere Objekte einem Prädikat zuordnen, mit einem Semikolon lassen sich mehrere Prädikate mit Objekt einem Subjekt zuordnen. Beispiel:

```
1 :Germany :label "Deutschland"@de ,
2           "Germany"@en ;
3           :populationTotal 81.799.600.000^^xsd:integer .
```

Listing 1: Beispiel RDF Deutschland

Dieses Beispiel zeigt auch, dass in RDF verschiedene Sprachen (siehe Zeile 1 und 2) hinterlegt werden können. In N3 kann dem entsprechenden Text einfach ein Sprachtag² angehängt werden.

Bislang wurde nur betrachtet, wie sich mit RDF Informationen zu Ressourcen zuordnen lassen können. Ein wichtiger Teil von semantischen Daten, nämlich die Semantik, haben wir bislang nicht betrachtet. Damit der Computer den Sinn vorhandener Daten verstehen kann, muss dieser noch Hintergrundinformationen [Hitzler et al., 2007, S. 66] kennen. Ressourcen sind in der Regel von einem bestimmten Typ (z.B. eine Stadt oder eine Person). Einer Person beispielsweise kann immer ein Vor- und ein Nachname zugewiesen werden. Diese Informationen lassen sich mittels Ontologien festlegen. Ontologien beschreiben eine bestimmte Domäne auf einem höheren Abstraktionslevel. Mit ihnen lassen sich bestimmte Zusammenhänge zwischen Ressourcen eines bestimmten Typs abbilden. Sie bilden ein allgemeines Vokabular, das zur Definition von Objekten einer Domäne benötigt wird. Dazu werden Methoden aus lexikalischer (also die Bedeutung durch Beziehungen), intensionaler (die Bedeutung durch Eigenschaften) und extensionaler (die Bedeutung durch die Instanzmenge) Semantik [Paulheim, 2011] verwendet. Dieses Hintergrundwissen ermöglicht dem Computer Aussagen zu verstehen und zu verarbeiten.

1.4 DBpedia

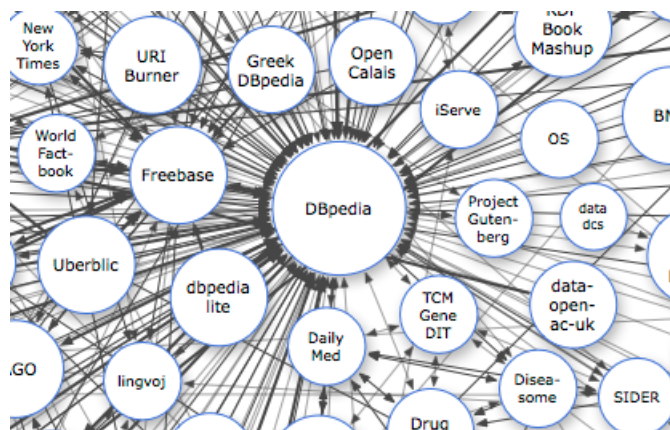


Abbildung 1: Verlinkungen von und zu DBpedia. Quelle: [Cyganiak and Jentzsch, 2011].

DBpedia ist eine der großen Datenquellen im Linked Open Data. Sie enthält Datensätze zu 3,77 Millionen Objekten, von denen 2,35 Millionen mit einer konsistenten Ontologie beschrieben werden³. *DBpedia* versucht automatisiert semantische Datensätze aus Wikipedia zu extrahieren und stellt diese maschinenlesbar bereit. Dazu werden Abbildungen zwischen den Daten in Wikipedia und der *DBpedia* Ontologie von Hand erzeugt [Auer et al., 2007, Jentzsch, 2009], mit dessen Hilfe die Datensätze automatisiert extrahiert werden können. Datenquellen die automatisiert generiert werden, haben den Nachteil teilweise veraltete oder falsche Daten zu enthalten. Die Datenqualität in *DBpedia* ist dennoch recht hoch, denn es

² Die Sprachtags werden nach dem ISO 963 Standard vergeben.

³ Stand August 2012

werden hauptsächlich die Informationsboxen von Wikipedia verwendet, um Daten zu extrahieren und semantisch aufzubereiten. Damit die Aktualität von DBpedia gegeben ist, werden die Wikipedia Artikel ständig überwacht und Änderungen in die Wissensdatenbank von DBpedia übertragen.

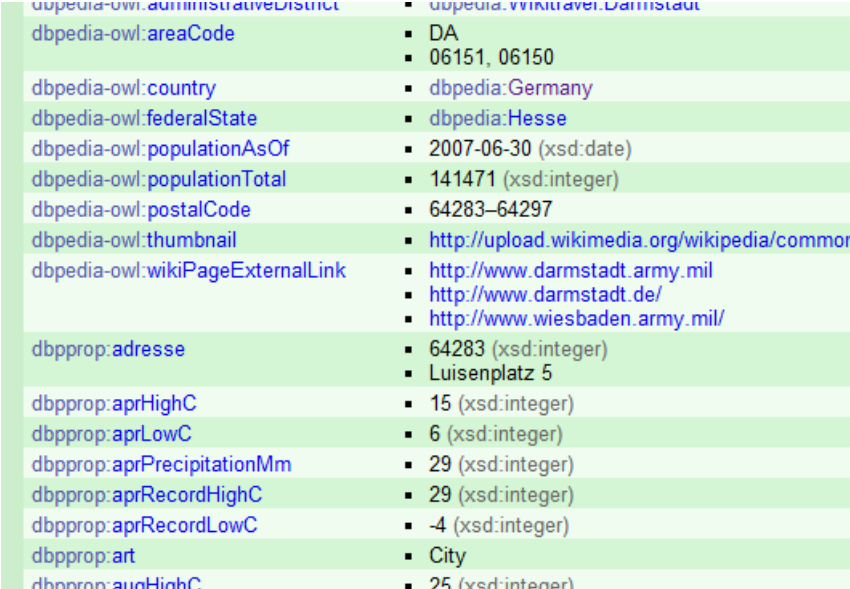
DBpedia hat im Linked Open Data eine große Bedeutung, denn viele andere Datenquellen zeigen auf Informationen in DBpedia und umgekehrt. Anhand der Abbildung 1 sieht man, dass es sehr viele eingehende Kanten von anderen Datenquellen zu DBpedia gibt. DBpedia ist bereits jetzt zu einer der wichtigsten Datenquellen im Linked Open Data geworden. Aber auch DBpedia nutzt externe Quellen für Verlinkungen. So verweist DBpedia zum Beispiel auf Bilder, externe Webseiten, externe RDF Datensätze, Wikipedia Kategorien und YAGO Kategorien [Auer et al., 2007].

2 Existierende semantische Browser für Linked Open Data

Linked Open Data können auf verschiedene Arten für den Benutzer dargestellt werden. In diesem Abschnitt sollen mehrere Browser vorgestellt werden, die eine Anzeige von Linked Open Data ermöglichen. Einige Datenquellen besitzen eigene Ansichten zur Darstellung von Informationen. Auch diese sollen hier berücksichtigt werden. In der hier durchgeführten Analyse sollen die Browser insbesondere auf Benutzerfreundlichkeit für Endbenutzer hin untersucht werden.

2.1 DBpedia Webzugriff

DBpedia bietet über die Webseite die Möglichkeit an, die dort hinterlegten Daten mittels eines Webbrowser anzuzeigen. Da DBpedia keine graphische Benutzeroberfläche für eine Suche besitzt, können Datensätze nur durch Eingabe des entsprechenden URI der Ressource aufgerufen werden. Als Ergebnis liefert DBpedia eine Anzeige aller verfügbaren Fakten zu der aufgerufenen Ressource. Die Darstellung beschränkt sich hauptsächlich auf die Anzeige der Eigenschaften mit ihren Werten in einer Tabelle. Die Tabelle ist der Eigenschaft nach alphabetisch sortiert und es werden alle Werte einer Eigenschaft in jeder verfügbaren Sprache angezeigt.



<code>dbpedia-owl:administrativeDistrict</code>	<code>dbpedia:wikitravel:Darmstadt</code>
<code>dbpedia-owl:areaCode</code>	<ul style="list-style-type: none">DA06151, 06150
<code>dbpedia-owl:country</code>	<code>dbpedia:Germany</code>
<code>dbpedia-owl:federalState</code>	<code>dbpedia:Hesse</code>
<code>dbpedia-owl:populationAsOf</code>	2007-06-30 (xsd:date)
<code>dbpedia-owl:populationTotal</code>	141471 (xsd:integer)
<code>dbpedia-owl:postalCode</code>	64283–64297
<code>dbpedia-owl:thumbnail</code>	http://upload.wikimedia.org/wikipedia/common
<code>dbpedia-owl:wikiPageExternalLink</code>	<ul style="list-style-type: none">http://www.darmstadt.army.milhttp://www.darmstadt.de/http://www.wiesbaden.army.mil/
<code>dbpprop:adresse</code>	<ul style="list-style-type: none">64283 (xsd:integer)Luisenplatz 5
<code>dbpprop:aprHighC</code>	15 (xsd:integer)
<code>dbpprop:aprLowC</code>	6 (xsd:integer)
<code>dbpprop:aprPrecipitationMm</code>	29 (xsd:integer)
<code>dbpprop:aprRecordHighC</code>	29 (xsd:integer)
<code>dbpprop:aprRecordLowC</code>	-4 (xsd:integer)
<code>dbpprop:art</code>	City
<code>dbpprop:ausHinhC</code>	25 (xsd:integer)

Abbildung 2: DBpedia-Ansicht zum Eintrag Darmstadt.

Wie man in der Abbildung 2 sehen kann, besitzen die Eigenschaften in der Anzeige keine natürlichen Namen. Die Eigenschaften werden mit ihren internen RDF Namen angezeigt und sortiert. Die angezeigten Eigenschaftsnamen sind für Entwickler wichtig, für den normalen Benutzer jedoch wenig benutzerfreundlich.

Da DBpedia vollständig automatisiert generiert wird, ist diese Art der Ansicht nicht verwunderlich. Es ist klar, dass diese Ansicht nie für den Endbenutzer entwickelt wurde, sondern dient zur Nutzung für Entwickler.

2.2 Disco - Hyperdata Browser

Der *Disco - Hyperdata Browser*⁴ [Bizer and Gauß, 2007] wurde von der Freien Universität Berlin entwickelt und ermöglicht das Anzeigen von beliebigen RDF Dokumenten. Dazu werden ähnlich der DBpedia

⁴ <http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/>

Ansicht die Prädikate mit deren Wert tabellarisch angezeigt. Zu den Prädikaten wird zudem angezeigt, aus welcher Quelle die Information stammt. Durch Auswahl der verknüpften Ressourcen können diese angezeigt werden. Eine Besonderheit von Disco ist, dass verknüpfte Bilder direkt in der Tabelle angezeigt werden. Die Prädikate lassen sich weder sortieren, noch sind diese in einer erkennbaren Reihenfolge angeordnet.

2.3 Tabulator

*Tabulator*⁵ [Berners-Lee et al., 2006] ist ein semantischer Browser, der beliebige RDF Dokumente anzeigen kann. Die Anzeige der Daten wird mittels einer Baumstruktur realisiert. Der Benutzer sieht eine Auflistung aller Prädikate der Ressource. Wenn ein Prädikat auf eine Verknüpfung zu einer anderen Ressource zeigt, kann der Benutzer diese in der gleichen Ansicht aufklappen und deren Informationen sehen. Die Auflistung der Prädikate folgt keiner besonderen Sortierung, sondern sie werden in der Reihenfolge aufgelistet, wie sie im RDF Dokument abgespeichert sind. Im Gegensatz zu der Ansicht von DBpedia und Disco nutzt der Tabulator Browser `rdf:label`⁶, um die Namen der Prädikate lesbar anzuzeigen. Es existieren noch zwei besondere Ansichten, die die Anzeige der Daten erleichtern: die Kalender- und die Kartenansicht. Sie ermöglichen eine komfortable Darstellung der Daten, die mit einem Datum bzw. mit geographischen Angaben versehen sind. Eine Besonderheit von Tabulator ist, dass RDF Dokumente mittels des Browsers editiert und gespeichert werden können.

2.4 aemoo

Ein semantischer Browser, der speziell die Verknüpfungen zwischen Ressourcen zeigt, ist der *aemoo*-Browser⁷. Er zeigt zu einer eingegebenen Ressource die wichtigsten Verknüpfungen zu anderen Ressourcen in einem Stern-Graph an. Die Anordnung und Auswahl der angezeigten Knoten werden durch Wissensmuster aus Wikipedia bestimmt, die mittels einer Benutzerstudie [Nuzzolese et al., 2011] herausgefunden wurden.

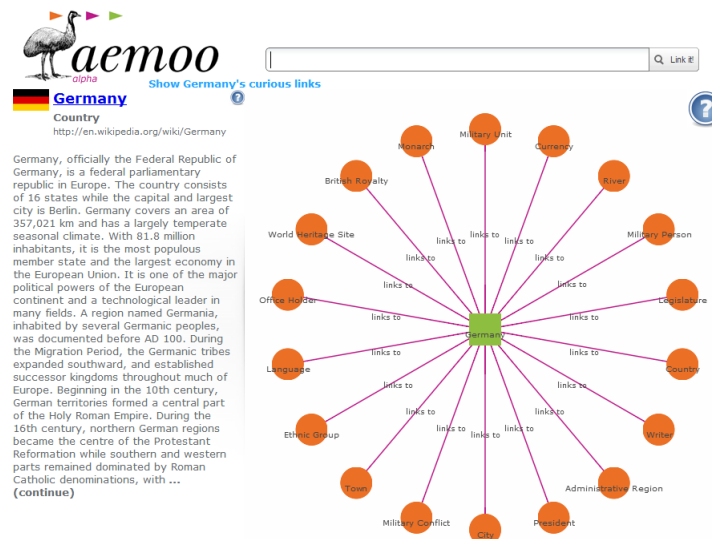


Abbildung 3: aemoo Ansicht von der Ressource Deutschland.

⁵ <http://www.w3.org/2005/ajar/tab>

⁶ `rdf:label` enthält eine kurze Beschreibung des Prädikats in verständlicher Sprache.

⁷ <http://wit.istc.cnr.it/aemoo/>

Nach Auswahl einer anderen Ressource wird die Ansicht aktualisiert und zeigt deren Verknüpfungen zu anderen Ressourcen an. Außer der Beschreibung links neben dem Graphen zeigt der Browser keine weiteren Eigenschaften der Ressource.

2.5 Freebase Webzugriff

Freebase⁸, eine weitere Linked Open Data Quelle, ähnelt DBpedia und enthält viele Artikel zu den unterschiedlichsten Themenbereichen. Bei Freebase werden die Datensätze allerdings nicht automatisiert gesammelt und erzeugt, sondern von Nutzern manuell eingepflegt. Hier kann sich der Nutzer mittels einer Suche bestimmte Informationen zu einer Ressource anzeigen lassen. Die Darstellung von Datensätzen unterscheidet sich aber zu DBpedia erheblich. Anstatt alle gefundenen Datensätze alphabetisch anzuzeigen, wird im oberen Teil der Anzeige eine kurze Zusammenfassung der aktuell betrachteten Ressource angezeigt. Anschließend werden Eigenschaften sinngemäß gruppiert und in Tabellenform dargestellt. Es finden sich allerdings nicht nur Tabellen in der Ansicht, sondern auch Graphen und Bilder.



Abbildung 4: Die Freebase-Ansicht zum Eintrag Darmstadt.

Im Vergleich zur DBpedia Ansicht ist die Ansicht bei Freebase deutlich übersichtlicher und benutzerfreundlicher, da Datensätze als Gruppe angezeigt werden. Jede Gruppe besitzt eine Überschrift, die den Inhalt der Gruppe beschreibt. Damit kann man leicht durch den Artikel zur gesuchten Information navigieren. Zudem werden Listen die viele Elemente enthalten gekürzt, um die Übersichtlichkeit zu erhöhen. Die Gesamtliste kann sich der Benutzer durch Klicken auf den entsprechenden Textlink anzeigen lassen.

Insgesamt wirkt die Ansicht viel durchdachter und sinnvoller. Diese Art der Anzeige lässt sich nur dadurch ermöglichen, dass Nutzer bei der Eingabe der Daten und bei der Gestaltung des Layouts von Ressourcen mithelfen. Da man bei Freebase bereits bei der Dateneingabe auf die Hilfe der Nutzer setzt, kann der

⁸ <http://www.freebase.com/>

Autor die Anzeige der neu eingegebenen Daten korrekt einsortieren, sodass eine in sich stimmende Ansicht entsteht.

2.6 Zusammenfassung

Die Mehrzahl der hier untersuchten semantischen Browser beschränkt sich bei der Ansicht der Datensätze auf eine Tabelle, die alphabetisch oder ungeordnet ist. Im Gegensatz zu Tabulator verzichten DBpedia und Disco bei der Darstellung der Eigenschaften sogar auf eine für den Menschen lesbare Beschriftung. Der aemoo Browser zeigt Eigenschaften von Ressourcen nicht an, sondern beschränkt sich auf die Verknüpfungen zu anderen Ressourcen. Die optisch übersichtlichste Darstellung der Datensätze ist in Freebase umgesetzt. Hier erfolgt die Anzeige der Eigenschaften in Gruppen, die zuvor manuell erstellt worden sind.

Bei dieser Untersuchung zeigt sich, dass es keinen Browser gibt, der sich die Semantik der Daten zunutze macht, um eine übersichtliche Darstellung von Daten zu ermöglichen. Die Gruppierung von Eigenschaften in der Freebase-Ansicht ist ein erster Schritt zu einer benutzerfreundlichen Ansicht. Eine automatisierte Gruppenfindung mit Hilfe der im Linked Open Data bereitgestellten Daten setzt kein derzeit verfügbarer Browser um.

3 Entwicklung eines semantischen Browsers

Im vorherigen Abschnitt haben wir eine Bestandsaufnahme existierender Lösungen zur Darstellung von semantischen Datensätzen durchgeführt. Die Darstellung von Freebase hat gegenüber anderen semantischen Browsern Vorteile für den Endbenutzer, insbesondere deren Übersichtlichkeit und Benutzerfreundlichkeit. Das Problem ist jedoch, dass die Gruppierungen dieser Datensätze von Menschen manuell vorgenommen werden müssen. Eine manuelle Bearbeitung der existierenden Datensätze aus automatisierten Datenquellen wie DBpedia macht aufgrund ihrer enormen Menge keinen Sinn. Es muss also eine Möglichkeit gefunden werden, diese Daten maschinell aufzubereiten.

3.1 Konzept der automatisierten Gruppierung

Der Grundansatz von Freebase, nämlich die Aufteilung der Datensätze in einzelne Gruppen, ermöglicht eine sehr übersichtliche Darstellung von Informationen. Ziel muss es also sein, eine automatisierte Möglichkeit zu finden, Gruppierungen für vorhandene Datensätze zu erzeugen, ohne dass Datensätze manuell bearbeitet werden müssen. Den Gruppen sollte anschließend eine Überschrift zugeordnet werden, der den Inhalt dieser kurz umschreibt. In dieser Arbeit sollen daher verschiedene Ansätze untersucht werden, wie automatisiert Gruppierungen von vorhandenen Datensätzen gefunden werden können.

Bei der Entwicklung dieser Methoden werden wir uns in dieser Arbeit auf eine Datenquelle, nämlich DBpedia beschränken. DBpedia besitzt eine große Anzahl an Ressourcen und liefert allgemeines Wissen zu unterschiedlichen Themenbereichen. Somit können die entwickelten Methoden durch die Anwendung dieser auf Ressourcen diverser Domänen getestet und evaluiert werden. Die im nachfolgenden vorgestellten Ansätze lassen sich jedoch auch auf beliebige andere Datenquellen anwenden. Sie sind nicht abhängig von einer speziellen Funktion oder Eigenschaft von DBpedia und lassen sich verallgemeinern.

3.2 Datensätze in RDF

Um zunächst einen Überblick zu gewinnen, welche Daten in RDF benutzt werden können, um Gruppierungen zu finden und um zu verstehen, welche Daten in RDF gespeichert sind, schauen wir uns zunächst die Datensätze an. Als Erläuterung wird dazu ein Auszug der Ressource Deutschland aus DBpedia herangezogen:

```
1 :Germany :populationTotal 81.799.600.000^^xsd:integer ;
2           :populationDensity 229.0^^xsd:integer ;
3           :label "Deutschland"^^xsd:string ;
4           :abstract "Deutschland ist ein föderalistischer Staat in Mi..."^^xsd:string .
5 :Darmstadt :country :Germany .
6 :Frankfurt :country :Germany .
```

Listing 2: Beispiel RDF Deutschland

Die Datensätze von Linked Open Data sind in RDF kodiert und bestehen somit immer aus drei Werten: Subjekt, Prädikat und Objekt. In RDF können Informationen in zwei unterschiedlichen Arten beschrieben werden:

- **Direkte Eigenschaft:**

Es wird eine Eigenschaft einer bestimmten Ressource direkt beschrieben. Hierbei ist die Ressource stets Subjekt und wird mit Prädikat und Objekt beschrieben. Beispiel:

```
:Germany :populationTotal 81.799.600.000^^xsd:integer .
```

Deutschland ist hier die Ressource über die gesprochen wird und ist im Beispiel Subjekt. Die *Einwohnerzahl* ist Prädikat mit der Zahl *81 Millionen* als Objekt.

- **Indirekte Eigenschaft:**

Im zweiten Fall wird indirekt eine Eigenschaft einer bestimmten Ressource beschrieben. Hier ist die Ressource das Objekt. Beispiel:

```
:Darmstadt :country :Germany .
```

In diesem Fall ist *Deutschland* Objekt und *Darmstadt* Subjekt. Trotzdem wird indirekt eine Information über Deutschland beschrieben, nämlich, dass Darmstadt eine Stadt in Deutschland ist.

Beide Typen von Informationen müssen bei der Entwicklung eines semantischen Browsers berücksichtigt werden und sollen später dem Benutzer angezeigt werden.

Es stellt sich nun die Frage, aus welchen Werten sich Informationen für die Gruppierung der Datensätze ablesen lassen. Da der semantische Browser Informationen zu einer bestimmten Ressource anzeigen soll, ist das Subjekt bei direkten Eigenschaften bzw. das Objekt bei indirekten Eigenschaften jeweils bekannt und identisch. Daraus lassen sich also keine Schlüsse ziehen. Im Fall von direkten Eigenschaften können Objekte von beliebigem Typ sein oder auf andere Ressourcen zeigen. Das Objekt ist sozusagen der Wert von einer Eigenschaft, beschreibt jedoch nicht, um welche Eigenschaft es sich handelt. Bei indirekten Eigenschaften dagegen sind Subjekte immer weitere Ressourcen, die in Verbindung mit der betrachteten Ressource stehen. Um aus diesen Ressourcen Informationen und deren Zusammenhang zu erkennen, müssten wir also zusätzlich diese ebenfalls auswerten. Eine einfachere Variante ist die Auswertung des Prädikats. Ein Prädikat beschreibt eine bestimmte Information über die Eigenschaft einer Ressource und ist somit die gesuchte Information zum Erkennen von ähnlichen Eigenschaften.

Damit wir die Information, ob es sich um direkte oder indirekte Eigenschaften handelt, nicht verlieren, ist es sinnvoll diese Prädikate getrennt zu betrachten und auszuwerten. Dazu sehen wir uns folgendes Beispiel an, um die Unterschiede zu erkennen:

Beispiel:

- 1 :Germany :partOf :Europe .
- 2 :Darmstadt :partOf :Germany .

Listing 3: Beispiel unterschiedliche Bedeutung indirekter und direkter Eigenschaften

Beide Prädikate haben hier exakt den gleichen Identitätsnamen und sind nur durch Betrachtung dieser nicht zu unterscheiden. Sie haben jedoch zwei unterschiedliche Bedeutungen: In Zeile 1 wird eine direkte Eigenschaft von Deutschland beschrieben, nämlich dass Deutschland Teil von Europa ist. In Zeile 2, dass Darmstadt ein Teil von Deutschland ist. Würden wir hier beide Prädikate nicht separat betrachten, könnten wir z.B. keine getrennten Gruppen für *Lage von Deutschland* und *Städte in Deutschland* anlegen. Damit dies jedoch möglich ist, trennen wir die Prädikate der direkten und indirekten Eigenschaften.

In unserem Beispiel aus Listing 2 gibt es die Prädikate $P = \{\text{populationTotal, populationDensity, label, abstract, country}\}$, die gruppiert werden sollen. Eine mögliche semantische Gruppierung wäre beispielsweise $P_1 = \{\text{populationTotal, populationDensity}\}$, $P_2 = \{\text{label, abstract}\}$, $P_3 = \{\text{country}\}$. In diesem Vorschlag zur Gruppierung wurden Eigenschaften gruppiert, die dem Sinn nach zusammengehören. So ist es für den Menschen naheliegend, die Bevölkerungszahl und die Bevölkerungsdichte in einer Gruppe zusammenzufassen. Ebenso verhält es sich mit der Beschriftung und der Zusammenfassung. Das Prädikat Land passt hier in keine der anderen Gruppe und bildet daher eine eigene Gruppe. Eine Gruppierung

nach dem Sinn ist jedoch immer eine subjektive Einschätzung eines jeden Einzelnen, weshalb man nicht zwischen richtiger und falscher Gruppierung unterscheiden kann.

3.3 Semantische Äquivalenz von Wörtern

Doch wie kann ein Computer semantische Ähnlichkeiten von Wörtern finden? Der Computer kann keine Einschätzung zu Dingen geben, die er nicht kennt oder die ihm nicht durch eine verständliche Beschreibung beigebracht wurde. Damit der Computer semantische Ähnlichkeit verstehen kann, werden in dieser Arbeit zwei Ansätze betrachtet:

- Mittels eines Wörterbuchs soll die semantische Ähnlichkeit von Wörtern berechnet werden.
- Mittels Suchmaschinen soll aus den Trefferlisten die Ähnlichkeit von Wörtern ermittelt werden.

Beides sind grundverschiedene Ansätze, die aber im Prinzip darauf aufbauen, eine Metrik zwischen Wörtern zu finden. Diese Metrik gibt die semantische Nähe zwischen Wörtern zueinander an. Im nachfolgenden sollen diese Ansätze kurz erläutert werden.

3.3.1 WordNet

*WordNet*⁹ ist eine von der Universität Princeton für die englische Sprache entwickelte Datenbank, die semantische und lexikalische Beziehungen zwischen Wörtern enthält. Sie wurde hauptsächlich für die Nutzung durch Maschinen erzeugt und optimiert. Substantive, Verben, Adjektive und Adverbien sind in sogenannte *Synsets* gruppiert. Synsets sind Gruppen, die jeweils einen bestimmten semantischen Zusammenhang beschreiben.

Im WordNet sind Substantive in Taxonomien unterschiedlicher Bedeutung nach organisiert, in welcher jeder Knoten eine Menge von Synonymen (ein Synset) repräsentiert. Ein Wort, das mehrere Bedeutungen hat, taucht auch in mehrere Synsets auf. Diese Synsets besitzen bidirektionale Zeiger auf andere Synsets, um ihre semantische Relation zueinander zu beschreiben. Diese Zeiger beschreiben die Relation *ist-von-der-Art* (Schreibweise $\{...\}@ \rightarrow \{...\}$), welche in die Pfeilrichtung eine Generalisierung darstellt und in die umgekehrte Richtung eine Verfeinerung. Diese Relation ermöglicht es Begriffe hierarchisch anzuordnen, beispielsweise $\{bird\}@ \rightarrow \{animal, animate_being\}@ \rightarrow \{organism, life_form, living_thing\}$. Diese Art von Darstellung erzeugt eine Hierarchie, die von konkreten Begriffen bis hin zu generischen Begriffen reicht [Fellbaum and Miller, 1998, S. 25]. Die unterschiedlichen Synsets sind in 25 unterschiedliche eindeutige Hierarchien¹⁰ zugeordnet. Diese Hierarchien beschreiben nicht überlappende Domänen, jede mit ihrem eigenen Wortschatz.

Durch den hierarchischen Aufbau des WordNets können Distanzen zwischen Begriffen berechnet werden. Da im WordNet semantisch ähnliche Begriffe sich in den gleichen Synsets oder in der Hierarchie nahe beieinander befinden, kann die Pfadlänge zwischen zwei Begriffen als semantische Distanz verwendet werden. Wir können die Pfadlänge zwischen a und b mittels der Formel

$$sim_{ab} = \max[-\log(Np/2P)]$$

berechnen, wobei Np die Anzahl der Knoten im Pfad a nach b und D die maximale Tiefe in der Taxonomie ist [Fellbaum and Miller, 1998, S. 274]. Es lassen sich aber auch noch weitere Metriken nutzen,

⁹ Webseite: <http://wordnet.princeton.edu/>. Aktuelle Version 3.0

¹⁰ WordNet Version 1.5 basierend auf dem Buch [Fellbaum and Miller, 1998, S. 28].

um Distanzen im WordNet zu berechnen. Eine Auswahl dieser werden im Artikel [Budanitsky and Hirst, 2001] beschrieben und evaluiert.

Mittels WordNet lassen sich also die Distanzen zwischen zwei Worten ermitteln. Was ist aber, wenn ein Prädikat aus mehreren Wörtern besteht? WordNet kann nicht die Distanz zwischen einem Satz oder Teilsatz zu einem anderen berechnen. Wir können dieses Problem lösen, indem wir die Distanzen zwischen den Wörtern beider Prädikate paarweise berechnen. Daraus kann dann eine durchschnittliche oder eine minimale Distanz zwischen den Prädikaten berechnet werden. Im Abschnitt 4 wird auf diese Unterscheidung im Hinblick auf Qualität der erzeugten Gruppen gesondert eingegangen.

Eine weitere Einschränkung muss im Hinblick auf die Wortart getroffen werden. Da im WordNet Substantive, Verben, Adjektive und Adverbien auf unterschiedliche Art und Weise gespeichert werden, kann eine Distanz zwischen Wortart verschiedener Worte nicht berechnet werden. Dieses Problem lässt sich für Verben umgehen, indem man mittels des Wörterbuchs das Verb nominalisiert. Das dadurch entstandene Substantiv kann dann wiederum für die Distanzberechnung verwendet werden. Bei der Implementierung muss jedoch darauf geachtet werden, dass Wörter aus mehreren Wortarten stammen können. Beispielsweise kann das Wort *name* als Substantiv für den Namen eines Gegenstandes verwendet werden, aber es gibt auch das Verb *to name*, um Dinge zu benennen. Prüft man also zuvor auf die Wortart Verb, dann nimmt man Nominalisierungen vor, obwohl es sich bereits um ein Substantiv handelt. Deshalb wird in der Implementierung des Browsers auf *nicht Substantiv* getestet, um das Problem zu umgehen.

WordNet besitzt standardmäßig bereits die Möglichkeit, falsch geschriebene oder abgeleitete Wörter zu erkennen und diesen den richtigen Synsets zuzuordnen. Es besitzt jedoch kein Abkürzungswörterbuch, sodass Prädikate die aus Abkürzungen bestehen nicht zugeordnet werden können. Zu Verbesserung der Qualität der Zuordnung wurde deshalb bei der Implementierung ein Abkürzungswörterbuch vorangestellt, das zuvor alle bekannten Abkürzungen durch die ausgeschriebenen Begriffe ersetzt.

3.3.2 Google-Distanz

Eine weitere Möglichkeit zur Ermittlung von Ähnlichkeiten zwischen Wörtern ist die Benutzung von Suchmaschinen. Rudi L. Cilibrasi und Paul M.B. Vitányi beschreiben in ihrem wissenschaftlichen Artikel „The Google Similarity Distance“ [Cilibrasi and Vitányi, 2004] den Einsatz von Google zur Bestimmung von Distanzen zwischen Wörtern. Die Grundidee basiert auf der Annahme, dass Wörter, die auf der gleichen Webseite vorkommen, auch ähnlich zueinander sind.

Mittels eines kurzen Beispiels lässt sich dieser Zusammenhang einfach erklären: Gibt man in der Suchmaschine Google das Wort „Pferd“ ein, erhält man 46.700.000 Treffer. Der Begriff „Reiter“ liefert 12.200.000 Treffer. Gibt man nun beide Begriffe zusammen ein, erhält man 2.630.000 Treffer.¹¹ Das heißt, dass beide Begriffe in einer relativ großen Anzahl in Webseiten gemeinsam auftreten. Cilibrasi und Vitányi haben eine Formel entwickelt, die einen Wert zwischen 0 (die beiden Begriffe sind sehr ähnlich zueinander) und 1 (die beiden Begriffe sind in keiner Weise zueinander semantisch ähnlich) liefert, um die semantische Distanz zwischen Begriffen zu errechnen. Dazu werden die Anzahl der Treffer der einzelnen Wörter, die beiden Wörter gemeinsam und die insgesamt von Google indizierten Seiten einbezogen:

$$NGD(x, y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x, y)}{\log M - \min\{\log f(x), \log f(y)\}}$$

Die Formel konsumiert zwei Begriffe x und y und liefert einen Wert als semantische Distanz. Die Funktion $f(x)$ liefert die Anzahl an Treffer von Google zum Begriff x . Die Konstante M beschreibt die Anzahl

¹¹ Beispiel aus dem Artikel „The Google Similarity Distance“ [Cilibrasi and Vitányi, 2004].

der von Google indizierten Seiten. Aus dem Beispiel oben in die Formel eingesetzt erhält man eine Distanz von

$$NGD(Pferd, Reiter) \approx 0.443$$

Mittels der Google-Distanz erhält man nun also, ähnlich zum WordNet Ansatz, ein Maß für die semantische Äquivalenz von Wörtern und Begriffen. Hierdurch kann man recht einfach das Problem von mehreren Wörtern in einem Prädikat lösen, indem man alle Wörter des Prädikats in die Suchmaschine eingibt. Im Gegensatz zum WordNet Ansatz muss auch nicht zwischen verschiedenen Wortarten unterschieden werden, da zur Berechnung der Distanz die Häufigkeit des Auftretens auf Webseiten entscheidend ist.

In der Praxis zeigen sich allerdings auch Nachteile dieses Lösungsweges. Google (aber auch andere Suchmaschinen für die der Ansatz ebenfalls einsetzbar¹² ist) beschränkt den Zugriff auf die Suchergebnisse auf eine maximal nutzbare Anzahl an Suchanfragen pro Tag und Monat. Da aber für jedes Wortpaar mehrere Anfragen gestellt werden müssen, wird die Begrenzung sehr schnell erreicht. Ein weiterer Nachteil ist die Geschwindigkeit. Trotz der recht schnellen Beantwortung von Suchanfragen benötigt das Programm mehrere Sekunden bis Minuten um Ergebnisse zu liefern. Das Problem hierbei ist die Masse der Anfragen, die gestellt werden muss. Für jede Distanzberechnung eines Wortpaares sind drei Suchanfragen notwendig. Der Nutzer des semantischen Browsers sollte aber nach Möglichkeit nicht lange warten, bis die entsprechenden Informationen angezeigt werden. Mittels Parallelisierung der Anfragen an die Suchmaschine ließe sich die Wartezeit vermutlich beschleunigen¹³. Trotzdem ist aus beiden genannten Gründen die Distanzberechnung mittels einer Suchmaschine aus Praxissicht nur sehr eingeschränkt nutzbar. Sie liefert aber bei den wenigen durchgeführten Tests ähnlich gute Ergebnisse wie der WordNet Ansatz.

3.4 Clustering

Wir haben nun im vorhergehenden Abschnitt gesehen, wie wir Distanzen zwischen Wörtern erhalten können. Wir suchen nun eine Möglichkeit, anhand der Beziehungen zwischen den Wörtern, Gruppen für die Eigenschaften zu bilden. Mittels eines Clustering-Algorithmus lassen sich aus den berechneten Distanzen Gruppen bestimmen. Ein Clustering-Algorithmus kann Ähnlichkeiten durch Analyse der Struktur von Daten erkennen und diese in Gruppen (genannt *Cluster*) zuordnen.

Je nach Clustering-Algorithmus ist der Gruppenbegriff unterschiedlich definiert. Für unsere Zwecke möchten wir Clustering Algorithmen verwenden, die eine eindeutige Klassifizierung eines Datensatzes in einem Cluster vornehmen. Ein Datensatz soll nicht zwei Clustern angehören. Wir wollen, dass der Clustering Algorithmus disjunkte Gruppen erzeugt. Dazu werden wir uns zwei unterschiedliche Clustering Ansätze ansehen, die diese Voraussetzungen erfüllen.

3.4.1 K-Means Clustering

Ein klassischer Clustering-Algorithmus ist der K-Means Algorithmus. Zuerst wird festgelegt wie viele Cluster erzeugt werden sollen. Diesen Parameter bezeichnet man mit k . Nun werden zufällig k Datensätze aus der Menge als Zentren gewählt. Alle Datensätze werden zu ihrem distanzmäßig nächsten Zentrum zugeordnet. Dazu wird die Distanz zwischen den Zentren und den anderen Datensätzen berechnet. Beispielsweise kann der euklidische Abstand verwendet werden. Als nächstes werden die Zentren neu berechnet, indem aus jeder Gruppe der Datensatz ausgewählt wird, zu welchem der Abstand zu

¹² Getestet wurde hauptsächlich auch die Suchmaschine *Bing* von Microsoft, die ebenfalls die gleichen Einschränkungen wie Google besitzt. Ab August 2012 wurde die maximale Anzahl der Anfragen pro Monat auf 5000 beschränkt.

¹³ Dies wurde jedoch aufgrund der genannten Beschränkungen der Anfragezahl nicht umgesetzt oder getestet.

allen anderen in dem Cluster befindlichen Datensätze am kleinsten ist. Der Vorgang so lange wiederholt, bis keine Datensätze mehr zwischen den Clustern wechseln [Witten et al., 2005, S. 137].

Der K-Means Clustering Algorithmus lässt sich recht einfach für die Verwendung anderer Distanzmetriken anpassen. Anstatt den euklidischen Abstand zu verwenden, können wir diesen einfach durch eine der im vorangegangenen Abschnitt beschriebenen Metriken ersetzen.

Bleibt noch die Wahl eines geeigneten k für die Anzahl der zu erzeugenden Gruppen. Aus psychologischer Sicht kann ein Mensch sich nur 7 plus minus 2 Dinge gleichzeitig merken [Miller, 1956]. Es liegt also nahe nur 7 Gruppen durch den Algorithmus erzeugen zu lassen. Da wir, wie bereits erwähnt, eine Unterscheidung zwischen direkten und indirekten Eigenschaften (Abschnitt 3.2) vornehmen, entstehen schlussendlich maximal 14 Gruppen für die Anzeige.

3.4.2 Bottom-Up

Ein weiterer Clustering Algorithmus ist der Bottom-Up Clustering Algorithmus, auch Hierarchisches Agglomerative Clustering genannt. Zu Beginn werden alle N Datensätze in einen eigenen Cluster zugeordnet. Bei jedem Schritt von den $N - 1$ Schritten werden die beiden am nächsten zueinander liegenden Cluster zusammengefasst. Jeder Schritt eliminiert also einen Cluster. Dazu wird zwischen jeden Clustern der Abstand paarweise zueinander berechnet. Wenn mehrere Datensätze in einem Cluster zusammengefasst wurden, gibt es beim Bottom-Up Clustering mehrere Möglichkeiten die Distanz zwischen den Clustern zu berechnen. Beim *Single Linkage* wird die minimale Distanz zweier Datensätze aus den beiden Clustern herangezogen (nearest-neighbor). *Complete Linkage* nimmt die größte Distanz zwischen zwei Datensätzen aus beiden Clustern an. Der durchschnittliche Abstand zwischen den beiden Clustern wird beim *Average Linkage* angenommen [Hastie et al., 2009, S. 523 ff].

Mit diesem Ansatz entsteht ein Baum, der nun zum Schluss noch bei einer gewählten Höhe abgeschnitten werden muss. Die daraus resultierenden Teilbäume sind nun die einzelnen Cluster, die die zusammengefassten Datensätze enthalten. Beim Bottom-Up Clustering kann nicht direkt bestimmt werden, wie viele Cluster am Schluss entstehen sollen. Die Anzahl ist abhängig von der gewählten Höhe und der gewählten Distanzberechnung. Unter Umständen können wenige Gruppen mit vielen Einträgen entstehen oder viele Gruppen mit wenigen Einträgen.

Bei der Entwicklung des semantischen Browsers hat sich gezeigt, dass der Bottom-Up Clustering Algorithmus für unsere Zwecke am besten mit Complete Linkage funktioniert. Als Schnitt für den Baum hat sich der Wert 5 am geeignetsten gezeigt:

Baumhöhe	Ø Gruppenanzahl	Ø Gruppengröße	Median Gruppengr.
3	5,27	56,81	16,00
4	8,55	33,60	10,00
5	13,20	20,55	6,50
6	19,22	12,97	4,00
7	25,54	8,94	3,00

Tabelle 1: Abhängigkeit der Gruppenanzahl und -größe von der gewählten Baumhöhe.

Für die Ermittlung der geeigneten Baumhöhe wurden 110 zufällige Datensätze unterschiedlichen Umfangs ausgewählt. Es wurden jeweils die Gruppenanzahl und die Größe der Gruppen im Durchschnitt berechnet. Die jeweiligen Ergebnisse können aus der obigen Tabelle abgelesen werden. Der Wert 5 hat sich als geeignetster Wert herausgestellt, da hier im Schnitt 13 Gruppen erzeugt werden. Damit fallen etwa im Schnitt 7 Gruppen auf direkte bzw. indirekte Eigenschaften, wodurch wir die menschliche Fähigkeit sich 7 plus minus 2 Dinge gleichzeitig zu merken [Miller, 1956] berücksichtigt haben. Ebenso ist

die Gruppengröße nicht zu groß oder zu klein, um auch bei umfangreicheren Datensätzen den Überblick zu verlieren.

Die hohen Werte für die durchschnittliche Gruppengröße entstehen durch einzelne Datensätze, die sehr umfangreich sind. Der Median gibt hier eine deutlich bessere Abschätzung für die durchschnittliche Gruppengröße bei normal großen Datensätzen an.

3.5 Gruppentitel

Um dem Benutzer noch einen besseren Überblick über die Inhalte der Gruppe zu geben, sollen im nächsten Schritt die Gruppen Titel erhalten. Der Titel sollte eine grobe Zusammenfassung der Datensätze geben. Der Clustering Algorithmus liefert keine Interpretationen über die gefundenen Gruppen, sodass hier eine weitere Analyse notwendig ist.

Eine Möglichkeit zur Erzeugung von Gruppentiteln ist die erneute Nutzung des WordNets. Durch die besondere Anordnung und Gruppierung der Worte im WordNet in Synsets, lässt sich zu zwei Worten ein gemeinsamer Vorgänger ermitteln. Dieser Vorgänger beschreibt eine Generalisierung der beiden Worte und ist demnach ein idealer Kandidat für einen Gruppentitel. Da jedoch in einer Gruppe mehrere Prädikate zusammengefasst wurden, muss ein gemeinsamer Vorgänger aller Prädikate gefunden werden. Durch paarweises Ermitteln der Vorgänger im WordNet kann ein Oberbegriff für alle Datensätze in einer Gruppe gefunden werden, der dann als Gruppentitel dient.

Beispiel:

Betrachten wir die Prädikate $P = \{\text{city, country, place}\}$. In diesem Fall würden wir zunächst den Vorgänger von city und country berechnen. Nehmen wir an, WordNet liefert uns $\{\text{city, country}\}@ \rightarrow \{\text{location}\}$, dann erhalten wir im nächsten Schritt folgende Berechnung $\{\text{location, place}\}@ \rightarrow \{\text{location}\}$. Das Wort location wird in diesem Fall als Gruppentitel verwendet.

Wurden jedoch Prädikate zusammen gruppiert, die im WordNet keine Relation zueinander haben (das kann immer dann passieren, wenn Synsets aus verschiedenen Hierarchiebäumen verglichen werden), kann WordNet keine aussagekräftigen Ergebnisse mehr liefern.

Eine weitere Lösung ist die Anwendung einer String-Analyse. Im einfachsten Fall wählt man das Prädikat aus den Datensätzen aus, das am meisten vorkommt und setzt es als Titel. Hierbei kann man allerdings keine Generalisierungen vornehmen, weshalb der Titel somit nur einen Teil der Eigenschaften einer Gruppe beschreiben kann.

In der Praxis hat sich gezeigt, dass eine Kombination beider Ansätze bessere Ergebnisse liefert. Als Primäransatz dient der WordNet-Ansatz, da dieser im Gegensatz zum textbasiertem Ansatz allgemeinere und sinnvollere Begriffe findet, um die Eigenschaften einer Gruppe zu umschreiben. Findet WordNet keine passenden Gruppennamen, wird auf die String-Analyse zurückgegriffen. Damit ist sichergestellt, dass jede Gruppe mindestens einen Titel besitzt, auch wenn dieser möglicherweise nicht alle Einträge repräsentiert.

3.6 Implementierung

Die in den vorangegangenen Abschnitten erläuterten Ansätze wurden im Rahmen dieser Bachelorarbeit in einen im Fachgebiet *Knowledge Engineering* der Technischen Universität Darmstadt entwickeltem Browser *Mob4LOD*¹⁴ implementiert. Dieser wurde parallel zu dieser Arbeit von einer Bachelorprakti-

¹⁴ <http://www.ke.tu-darmstadt.de/resources/mob4lod>

kumsgruppe¹⁵ entwickelt. Der Browser wurde mittels Java und dem *Grails Framework*¹⁶ für Webapplikationen umgesetzt und ermöglicht eine einfache Entwicklung von Filtern, die die Darstellung der Datensätze verändern können. So lassen sich Gruppen mit Gruppentitel erzeugen, sowie Datensätze filtern oder sortieren.

Für die Implementierung in den bestehenden Browser war lediglich die Entwicklung eines Filters (hier: *SemanticFilter*) notwendig, der die vorgestellten Ansätze implementiert. Das nachfolgende Klassendiagramm (Abbildung 6) zeigt den modularen Aufbau der Implementierung. Sowohl die Distanzfunktionen, die Generierung der Gruppentitel als auch die Clustering Algorithmen lassen sich beliebig austauschen.

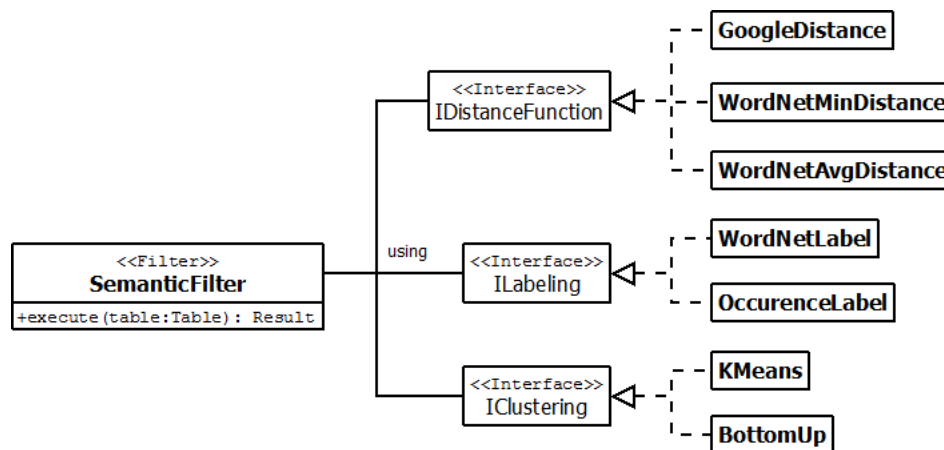


Abbildung 5: Aufbau der Implementierung des sematischen Browsers.

Die Klasse *Filter* ist eine abstrakte Klasse des Browsers, die überschrieben werden muss, um die Ausgabe zu beeinflussen. Sie enthält die Methode `execute`, die die Tripel des aufgerufenen RDF Dokuments bereitstellt und soll die verarbeiteten RDF Tripel zurückgeben. Durch die Manipulation der Tripel kann die schlussendliche Ausgabe beliebig angepasst werden. Es lassen sich Gruppen mit Tripeln oder weiteren Untergruppen erzeugen und beliebig verschachteln.

Mit der Filterklasse *SemanticFilter* wird der komplette Ablauf der Bildung von Gruppen implementiert. Dabei werden zunächst die übergebenen RDF Tripel in Tripel mit direkten und indirekten Eigenschaften zerlegt, um sie separat zu bearbeiten. Im Anschluss daran werden die Tripel dem ausgewählten Clustering Algorithmus (*IClustering*) mit der entsprechenden Distanzfunktion (*IDistanceFunction*) übergeben, damit dieser die Gruppenbildung vornehmen kann. Für die Clustering Algorithmen stehen die Klassen *KMeans* und *BottomUp* zur Verfügung. Wegen der unterschiedlichen Konzepte der beiden Clustering Algorithmen, benutzen diese zwei unterschiedlichen Datenrepräsentationen. So arbeitet K-Means auf einer Listenstruktur, während der Bottom-Up Algorithmus auf einer Baumstruktur arbeitet. Als Distanzfunktionen können die Klassen *WordNetAvgDistance* (berechnet die durchschnittliche Distanz von Wörtern mittels WordNet), *WordNetMinDistance* (berechnet die minimale Distanz von Wörtern), *GoogleDistance* (berechnet die Distanz mittels der Suchmaschine Google) und *BingDistance* (berechnet die Distanz mittels der Suchmaschine Microsoft Bing) verwendet werden. Zur Repräsentation von mehreren Wörtern in einem RDF Prädikat existiert die Klasse *WordList*. Sie speichert die Wörter getrennt, sodass die Distanzmetriken im Falle mehrerer Wörter einfachen Zugriff auf die getrennten Wörter besitzen.

Nachdem der erste Clusteringdurchlauf beendet ist, wird eine weitere Analyse der Daten gestartet. Besitzt eine Gruppe mehr als drei unterschiedliche Eigenschaften, dann wird ein zweiter Durchlauf durch-

¹⁵ Der Browser selbst ist nicht Teil dieser Arbeit oder wurde vom Autor erstellt. Er wurde um die in dieser Arbeit vorgestellten Ansätze erweitert.

¹⁶ Das Grails Framework (<http://www.grails.org>) ermöglicht eine einfache Entwicklung von Webapplikationen auf der Basis von Java.

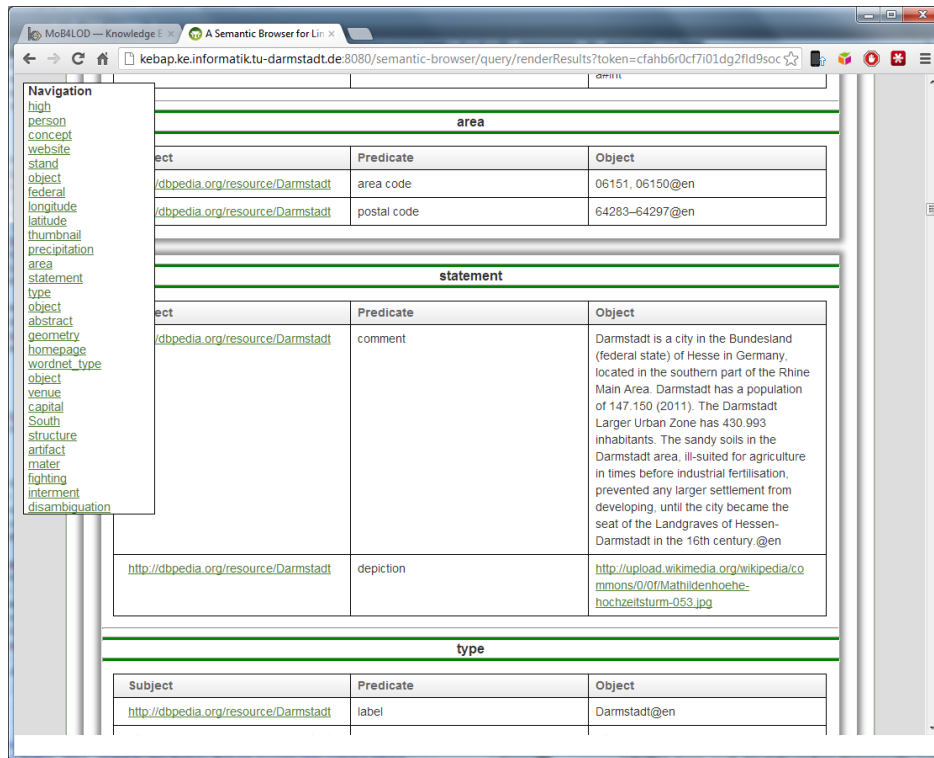


Abbildung 6: Webansicht des semantischen Browsers der Ressource *Darmstadt*. Zu sehen sind die erzeugten Gruppen, die Navigationsleiste (links) und die RDF-Tripel.

geführt. Dieser Durchlauf erzeugt mittels K-Means Clustering genau drei Untergruppen. Die Überlegung dabei war, größere Gruppen nochmals aufzuteilen, um die Übersicht zu erhöhen. Sind alle Gruppen erzeugt worden, dann werden diese aus der internen Darstellung in die des Browsers überführt. Abschließend wird den erzeugten Gruppen jeweils ein Gruppentitel mittels der ausgewählten Klasse (ILabeling) zugeordnet.

Zusätzlich zu den vorgestellten Klassen existieren Hilfsklassen für das Verarbeiten von Wörtern (beispielsweise das Trennen von Wörtern oder das Löschen von Stoppwörtern) und das Auflösen von Abkürzungen.

Damit in der Webansicht eine Anzeige der Gruppentitel neben den Daten platziert werden konnte, musste das Template des Browsers minimal verändert werden. Die Erzeugung dieser Ansicht wird ausschließlich über ein JavaScript mittels jQuery¹⁷ realisiert. Dadurch war es nicht notwendig, grundlegende Änderungen am Browserframework oder am Template vorzunehmen.

3.7 Optimierungen

Während der Entwicklung des semantischen Browsers hat sich gezeigt, dass die Antwortzeit von der Anfrage bis hin zur Anzeige zwischen 30 bis 40 Sekunden liegt. Damit der Browser effizient nutzbar bleibt, sollte die Antwortzeit nicht länger als 10 Sekunden betragen.

Bei der Analyse des Problems stellte sich heraus, dass die Abfrage der Beschreibungen der Eigenschaften eines der Hauptprobleme für die lange Antwortzeit ist. Dies liegt daran, dass für jede Eigenschaft eine separate Abfrage an die Datenquelle gestellt werden muss.

¹⁷ jQuery ist eine freie JavaScript Bibliothek, die diverse JavaScript Funktionen vereinfacht. Insbesondere HTML Traversierungen lassen sich sehr elegant lösen. Webseite: <http://www.jquery.com/>

Im Falle von DBpedia gibt es zwei Arten von Eigenschaften: präzise Definitionen innerhalb einer konsistenten Ontologie und weniger spezifische Eigenschaften. DBpedia bietet die Ontologie (inklusive der Beschreibungen der Eigenschaften) als auch die Beschreibungen für weniger spezifische Eigenschaften als separaten Download an. Damit können die Beschreibungen aller Eigenschaften lokal ermittelt werden, ohne dass ein Herunterladen jeder Einzelnen während der Erzeugung der Gruppen notwendig ist. Das Vorladen der Eigenschaften kann auch für jede andere Datenquelle realisiert werden. Eine weitaus bessere Lösung dieses Problems wäre den Browser näher an den Daten zu platzieren, sodass eine Zwischenspeicherung der Beschreibungen nicht mehr notwendig ist. Der Browser kann dann ohne Umwege über einen API-basierten Ansatz die Daten aus dem RDF Speicher auslesen.

Eine weitere Optimierung wurde am Clustering Algorithmus Bottom-Up vorgenommen. Anstatt die Distanzen zwischen den Prädikaten bei jedem Schritt neu zu berechnen, werden diese vor dem eigentlichen Algorithmus berechnet und zwischengespeichert. Dadurch sind Mehrfachberechnungen ausgeschlossen, die zusätzliche Zeit benötigen würden.

Wie bereits in Abschnitt 3.3 erwähnt, ist die Distanzberechnung mittels einer Suchmaschine sehr langsam und damit für den produktiven Einsatz wenig geeignet. Trotzdem lässt sich der Ansatz optimieren, um die Berechnung häufig verwendeter Prädikate zu beschleunigen. Dazu wurde ein Zwischenspeicher entwickelt, der die Anzahl der Suchtreffer speichert. Mit diesem Speicher müssen nur noch Anfragen zu Prädikaten an die Suchmaschine gestellt werden, deren Trefferanzahl noch nicht abgerufen wurde. Dies beschleunigt die Antwortzeit des Browsers allerdings nur bei Datenätzen, die öfter als einmal aufgerufen werden.

4 Evaluierung

Als Evaluation des semantischen Browsers wurde eine Benutzerstudie durchgeführt. Sie soll zeigen, ob mittels einer semantischen Gruppierung die Beantwortung von Fragen zu einer bestimmten Ressource schneller ist, als die mit den bislang verfügbaren Darstellungen.

Neben der Benutzerstudie wurden auch Performancetests durchgeführt. Mit Hilfe dieser Tests soll untersucht werden, ob der Browser in der Praxis sinnvoll einsetzbar ist. Relevant ist hierbei insbesondere die Zeit, die benötigt wird, um die Ansichten für Ressourcen zu berechnen und darzustellen.

4.1 Testinstanzen und -gruppen

In diesem Abschnitt werden die Testinstanzen vorgestellt, die bei der Benutzerstudie verwendet worden sind. Ebenso wird die Testgruppenverteilung erläutert.

4.1.1 Testinstanzen

Bei der Benutzerstudie wurden zufällig jeweils 10 Instanzen aus folgenden Themenbereichen aus DBpedia ausgewählt: Länder, Städte und Kinofilme. Bei der Auswahl wurde darauf geachtet, dass die Instanzen genügend RDF-Tripel (in englischer Sprache) besitzen, sodass eine Gruppierung für diese Ressourcen möglich und sinnvoll ist.

Zu jeder Instanz wurde eine Frage formuliert, die mittels der Fakten beantwortet werden kann. Dabei wurde darauf geachtet, dass die Fragen nicht direkt den Namen des Prädikats enthalten, damit einfaches Suchen nicht zur Lösung führt. Dies ist allerdings nicht in jedem Fall möglich, kann aber zeigen bei welchen Fragen eine Gruppierung der Eigenschaften sinnvoller ist, als bei anderen. Außerdem erfordern manche Fragen die Suche nach mehr als einem Prädikat, die semantisch ähnlich sind. Eine gute Gruppierung sollte dazu führen, dass diese Fragen schneller beantwortet können, da deren Prädikate möglichst in ein und derselben Gruppe sortiert wurden. Ebenso wurde darauf geachtet, dass Fragen nicht direkt aus dem Allgemeinwissen des Testers beantwortet werden können.

Die Evaluation wurde ausschließlich mit Instanzen durchgeführt, deren Datensätze in englischer Sprache verfügbar sind. Die meisten vorgestellten Ansätze sind derzeit nur für die Sprache Englisch einsetzbar, da die verwendeten Tools (beispielsweise WordNet) keine anderen Sprachen unterstützen. Alle Datensätze anderer Sprachen wurden daher zuvor herausgefiltert, um das Ergebnis nicht zu verfälschen.

Für jede der 30 Instanzen wurden drei verschiedene Ansichten erzeugt:

(A) Baseline

Die Baseline-Darstellung zeigt dem Benutzer die Liste der Fakten in alphabetischer Reihenfolge an. Die Fakten werden in 9 Gruppen eingeteilt, die jeweils drei Anfangsbuchstaben repräsentieren. Also A - C, D - F,

Da nicht alle Anfangsbuchstaben abgedeckt werden, ergibt sich durch das Ignorieren leerer Gruppen, folgende Verteilung:

Ø Anzahl Gruppen	Ø Gruppengröße
4,6	103,9

Tabelle 2: Gruppenanzahl und -größe bei Baseline

(B) K-Means mit WordNet

Die K-Means-Darstellung zeigt dem Benutzer die Fakten in semantischen Gruppen an, die mittels des K-Means-Clustering Algorithmus ($k = 7$) erzeugt wurden. Als Distanzmetrik für die Fakten wurde der WordNet Ansatz mit durchschnittlichem Abstand zwischen mehreren Werten herangezogen. Der durchschnittliche Abstand bei der Distanzberechnung hat sich in Kombination mit K-Means als die bessere Variante gegenüber der minimalen Distanz herausgestellt. Es wurde jedoch davon abgesehen, dies durch eine Evaluierung zu belegen. Die Abweichung der Gruppenanzahl und -größe zur minimalen Distanz sind jedoch nur gering. Zusätzlich werden vor der Distanzberechnung Abkürzungen ersetzt und Stoppwörter gelöscht.

Die angezeigten Ressourcen haben nach der Gruppierung die folgenden Eigenschaften:

Ø Anzahl Gruppen	Ø Gruppengröße
10,6	53

Tabelle 3: Gruppenanzahl und -größe bei K-Means

Die durchschnittliche Anzahl der Gruppen von 10,6 erklärt sich dadurch, dass leere Gruppen in die Berechnung nicht mit einfließen, da sie auch nicht angezeigt werden.

(C) Bottom-Up mit WordNet

Bei der Bottom-Up-Darstellung zeigt der Browser dem Benutzer die Fakten in semantischen Gruppen an. Diese wurden mittels des Bottom-Up Clustering Algorithmus (Baumhöhe $h = 5$) mit Complete Linkage erzeugt. Als Distanzmetrik wurde ebenfalls das WordNet verwendet, hier allerdings mit der minimalen Distanz zwischen den Wörtern. Die minimale Distanz hat sich in Kombination mit dem Bottom-Up Clustering Algorithmus als bessere Metrik herausgestellt. Auch hier wurden vor der Distanzberechnung die Abkürzungen ersetzt und Stoppwörter gelöscht.

Die daraus resultierenden Gruppierungen setzen sich wie folgt zusammen:

Ø Anzahl Gruppen	Ø Gruppengröße
15,7	35,3

Tabelle 4: Gruppenanzahl und -größe bei Bottom-Up

4.1.2 Testgruppe

Die Tester wurden ebenfalls in drei Testgruppen 1, 2 und 3 eingeteilt. Jeder Testgruppe wurden alle 30 Instanzen angezeigt. Dabei kann die Verteilung der Darstellungen der folgenden Tabelle entnommen werden.

Fragen	(A) Baseline	(B) K-Means	(C) Bottom-Up
1 - 10	2	1	3
11 - 20	3	2	1
21 - 30	1	3	2

Tabelle 5: Verteilung der Darstellungsformen bei der Evaluierung.

10 Personen haben die Evaluierung mittels des automatisierten Testprogramms durchgeführt. Von diesen Testpersonen sind 4 Personen mit dem Thema Linked Open Data vertraut gewesen oder haben bereits mit dem semantischen Web gearbeitet. Vier Tester wurden in die Gruppe 1, und jeweils drei Tester in die Gruppen 2 und 3 aufgeteilt.

4.2 Ablauf der Benutzerstudie

Dem Benutzer wird in einer Ansicht (Abbildung 7) die Instanz mit der dazugehörigen Frage angezeigt. Sobald die Darstellung der Fakten geladen wurde, wird die Zeit gestoppt, die benötigt wird, um die Frage zu beantworten. Der Benutzer kann die Antwort in ein Textfeld eingeben und bestätigt die Eingabe durch Klick auf „OK“.

Die Darstellungsansicht der jeweiligen Testgruppe wechselt alle 10 Fragen in die nächste Darstellungsform (vgl. Tabelle 5). Zuvor wird der Nutzer noch gefragt, wie ihm die bisherige Darstellung gefallen hat. Er kann dazu Schulnoten von 1 (sehr gut) bis 6 (ungenügend) vergeben. Nachdem der Nutzer alle 30 Fragen beantwortet hat, ist der Durchlauf beendet.

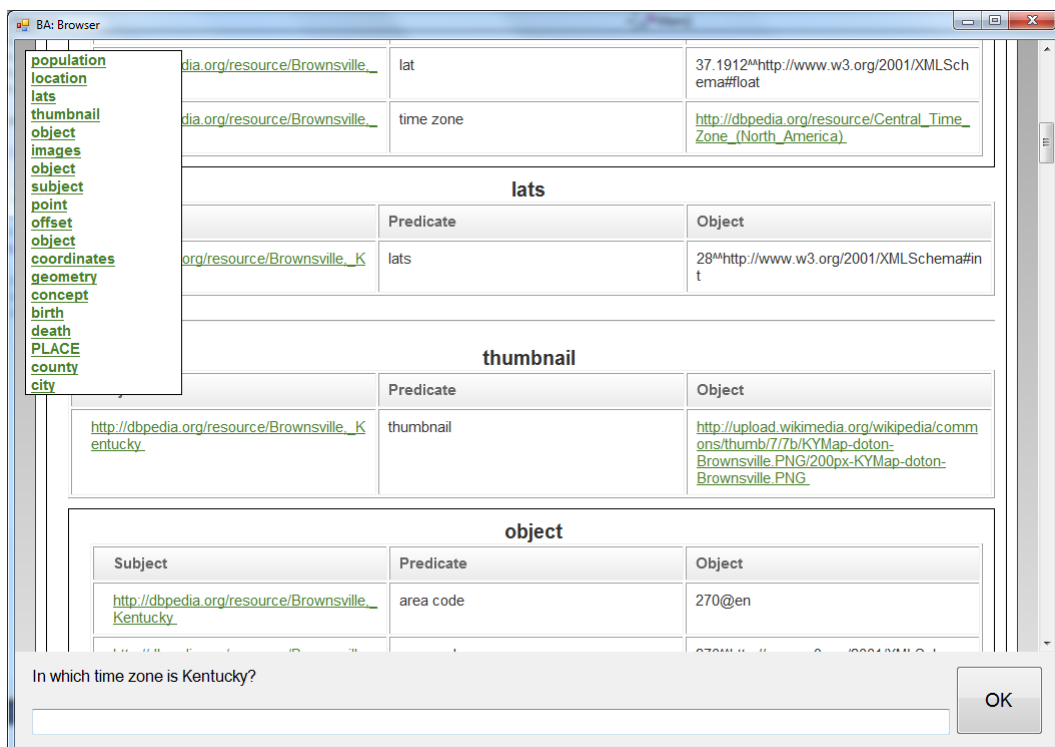


Abbildung 7: Evaluationsprogramm. Es zeigt die Ressource *Kentucky* in der *Bottom-Up* Ansicht.

In allen Darstellungsformen werden neben der Anzeige der Gruppen zusätzlich die Gruppentitel angezeigt. Diese zusätzliche Anzeige ermöglicht es schnell zwischen den Gruppen zu wechseln. Durch Anklicken auf einen Gruppentitel springt die Ansicht zu der entsprechenden Gruppe. Die Anzeige ist oben links angeordnet, sodass jederzeit der gesamte Umfang der angezeigten Ressource sichtbar bleibt. Im Testprogramm selbst kann keine Suche durchgeführt werden, ebenso ist das Kopieren von Textinhalten durch den Tester nicht möglich. Hierdurch soll für jeden Tester annähernd die gleichen Voraussetzungen geschaffen werden, um das Ergebnis der Benutzerstudie nicht zu beeinflussen.

4.3 Auswertung

Zunächst betrachten wir die Bewertung der Ansichten durch die Tester. Das Testprogramm hat die Tester nach jedem Wechsel der Ansicht über die vorhergehende befragt. Dabei sollten diese die Darstellung mit einer Schulnote bewerten. Die nachfolgende Tabelle zeigt die durchschnittliche Bewertung der drei verschiedenen Ansichten:

Rang	Ansicht	Ø Bewertung	Schlechteste	Beste
1.	(C) Bottom-Up	2,9	5	2
2.	(A) Baseline	3,0	5	2
3.	(B) K-Means	3,4	6	2

Tabelle 6: Bewertung der Ansichten durch die Tester.

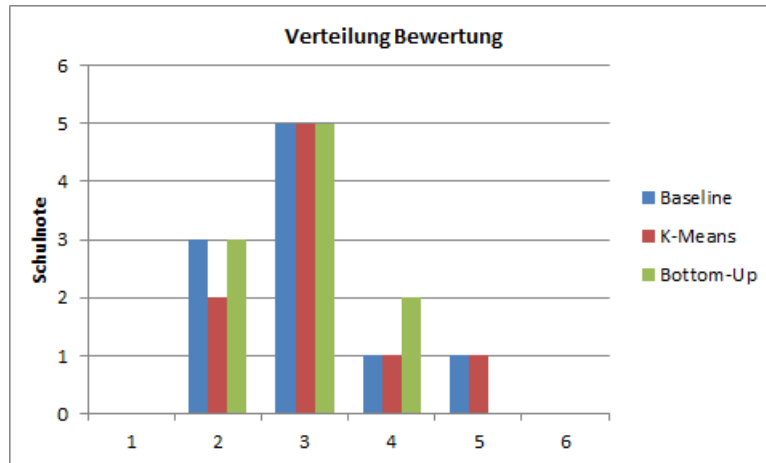


Abbildung 8: Verteilung der Benutzerbewertungen nach Ansicht.

Insgesamt haben die Tester die Ansichten sehr unterschiedlich beurteilt. Keine der Ansichten hat einen signifikanten Vorsprung gegenüber der Baseline und liegen deshalb relativ nah beieinander. (C) Bottom-Up wurde bei der Befragung als beste Ansicht mit 2,9 benotet, gefolgt von der (A) Baseline mit der Note 3,0 und schließlich der Ansicht mit (B) K-Means mit einer Bewertung von 3,4. Die schlechteste Bewertung von einer Person war die Note 6 (ungenügend) für K-Means. Trotz der schlechten Durchschnittsnoten wurde jede Ansicht mindestens einmal mit der Note 2 bewertet. Ein allgemeiner Trend, welche Ansicht subjektiv die bessere ist, kann man aufgrund dieses Ergebnisses nicht erkennen. Die Bewertungen sind nicht signifikant unterschiedlich, weshalb man keiner der hier getesteten Gruppenansichten ein besseres Ergebnis nachweisen kann. Der Signifikanztest (T-Test) zwischen der Baseline und K-Means liefert eine Wahrscheinlichkeit von 43%; zwischen Baseline und Bottom-Up 79%. Die Hypothese, dass die Messreihen mit einer Wahrscheinlichkeit von 95% signifikant unterschiedlich sind, muss in beiden Fällen abgelehnt werden.

Als weiteres Kriterium für die Evaluierung wurde die Zeit für das Beantworten von Fragen zu einer bestimmten Ressource gemessen. Die Annahme dahinter: Je schneller ein Tester die Frage beantworten kann, desto höher ist auch die Qualität der Anzeige. Da alle zufällig ausgewählten Ressourcen für alle Tester unbekannt waren, konnte sichergestellt werden, dass Tester die Frage nicht aus dem Allgemeinwissen beantworten konnten. Alle Tester waren somit darauf angewiesen, die Informationen aus der Ansicht der Ressource zu finden und mit diesen die Frage zu beantworten. Insgesamt wurden 300 Fragen und Antworten ausgewertet, wovon 15 Antworten nicht gezählt wurden, da sie von den Testern falsch beantwortet wurden. Die falschen Antworten sind auf alle drei Ansichten ungefähr gleich häufig verteilt (Bottom-Up = 5; K-Means = 7; Baseline = 3). Demnach wurden nur lediglich 5% der Fragen falsch beantwortet.

Bei der Auswertung der Messwerte stellte sich heraus, dass unbereinigt kein signifikanter Unterschied zwischen den verschiedenen Ansichten vorlag. Der T-Test¹⁸ lieferte Werte, bei denen die Hypothese ab-

¹⁸ Hypothese: beide Messreihen unterscheiden sich signifikant voneinander. Die Hypothese wird angenommen, falls die Wahrscheinlichkeit hierfür größer als 95 % ist.

gelehnt werden muss. Ein eindeutiges Ergebnis konnte daher nicht abgelesen werden. Da keinem der Tester vorab die Ansichten gezeigt wurden, muss man eine „Eingewöhnungsphase“ einplanen. Bei der Auswertung wurden daher jeweils die ersten beiden Fragen einer Ansicht gestrichen, wodurch pro Ansicht 8 Fragen übrig blieben. Der T-Test für die Baseline und K-Means lieferte eine Wahrscheinlichkeit von 0,000607 und ist damit hoch signifikant, der T-Test für die Baseline und dem Bottom-Up Ansatz lieferte eine Wahrscheinlichkeit von 0,053948 und ist demnach nur knapp nicht signifikant. Das heißt, dass ein messbarer Unterschied zwischen den Ansichten zur Baseline nachgewiesen werden konnte. Zur Bewertung welche Ansicht nun signifikant besser funktioniert, wurde die durchschnittliche Beantwortungsdauer betrachtet. Diese liegt für die oben genannten bereinigten Messwerte für die Baseline bei 34,8, für K-Means bei 50,4 und für Bottom-Up bei 43,4. Das heißt, dass mit der Baseline Ansicht eine schnellere Beantwortung der Fragen möglich ist, als mit einer der gruppenbasierten Ansichten.

Die Tabelle 7 enthält eine detaillierte Aufschlüsselung über die Ergebnisse der Benutzerstudie. Sie gibt die einzelnen Werte gruppiert nach den drei verschiedenen Ressourcenarten (Städte, Länder, Filme) an und zeigt die Ergebnisse der einzelnen Ansichten. Abzulesen ist die durchschnittliche Anzahl der erzeugten Gruppen, die durchschnittliche Gruppengröße, die durchschnittliche Berechnungsdauer für die Erzeugung der Gruppen und die durchschnittliche Beantwortungsdauer durch die Tester. Zusätzlich kann in Klammern der Median abgelesen werden, der „Ausreißer“ weniger stark berücksichtigt, als der Mittelwert. Die Tester beantworteten die Fragen insgesamt mittels der Baseline am schnellsten, gefolgt von Bottom-Up und K-Means. Ein ganz eindeutiges Bild ergibt sich hieraus nicht, da die Tester die Fragen zu Ländern schneller mit der K-Means Ansicht im Gegensatz zum Bottom-Up Ansatz beantworteten. Im Allgemeinen lässt sich jedoch ableiten, dass die Tester offensichtlich mit einer einfachen alphabetischen sortierten Liste schneller die Antwort auf die gestellte Frage finden, als mit einer gruppierten Ansicht.

Algorithmus	Baseline	K-Means	Bottom-Up
durchschnittliche (Median) Anzahl von Gruppen			
Städte	5,6 (4,0)	12,4 (13,0)	17,9 (17,5)
Länder	5,1 (5,5)	12,0 (12,0)	17,4 (15,5)
Filme	3,2 (3,0)	7,3 (7,3)	11,7 (12,0)
durchschnittliche (Median) Gruppengröße			
Städte	184,0 (37,0)	104,0 (12,5)	71,4 (8,0)
Länder	95,8 (22,5)	41,1 (8,0)	26,5 (6,5)
Filme	31,8 (30,5)	13,1 (12,5)	8,0 (8,0)
durchschnittliche (Median) Berechnungsdauer (in Sekunden)			
Städte	< 1 (< 1)	14,1 (4,6)	15,4 (5,1)
Länder	< 1 (< 1)	3,9 (1,6)	4,3 (1,7)
Filme	< 1 (< 1)	0,5 (0,4)	0,4 (0,4)
durchschnittliche (Median) Beantwortungsdauer Tester (in Sekunden)			
Städte	33,6 (25,1)	65,0 (51,6)	40,3 (36,1)
Länder	54,5 (40,0)	47,7 (42,7)	60,0 (51,1)
Filme	33,0 (31,1)	36,1 (27,4)	37,0 (32,0)

Tabelle 7: Ergebnis der Benutzerstudie. Quelle: [Seeliger and Paulheim, 2012]

Betrachtet man ausschließlich die durchschnittliche Beantwortungsdauer je Frage, dann werden im Durchschnitt etwa die Hälfte (17) der Fragen mit einer gruppierten Ansicht schneller beantwortet. 8 der 10 Fragen fallen auf die Kategorie Länder, 7 von 10 Fragen auf Filme und 4 von 10 auf Städte. Das bedeutet, dass der K-Means Ansatz für Länder eine bessere Ansicht produziert, als die Baseline oder der Bottom-Up Algorithmus. Für die Kategorie Städte liefert weder der K-Means noch der Bottom-Up Algorithmus besonders gute Ergebnisse für die Gruppenerzeugung. Grund hierfür dürften die Wetterdaten

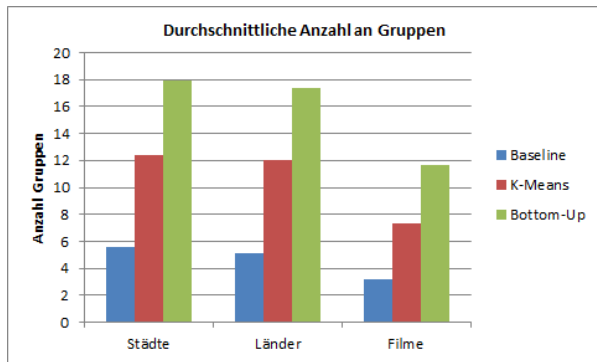


Abbildung 9: Durchschnittliche Anzahl an Gruppen, gruppiert nach Städten, Ländern und Filmen.

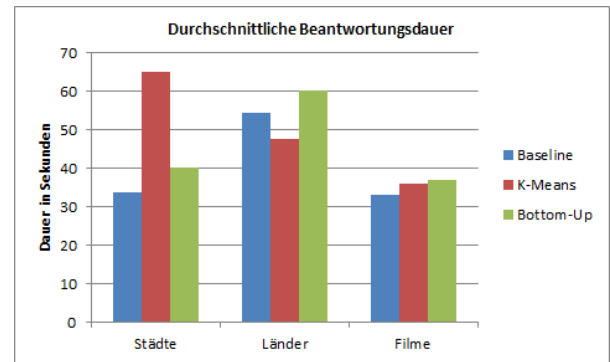


Abbildung 10: Durchschnittliche Beantwortungsdauer, gruppiert nach Städten, Ländern und Filmen.

(minimale, maximale und durchschnittliche Temperatur, und die Niederschlagsrate pro Monat) sein, die im Falle des K-Means zu einer Vermischung mit anderen Daten, und für Bottom-Up zu der Erzeugung von vielen kleinen Gruppen führte. Ein signifikanter Unterschied zwischen den Ansichten für die hier schneller beantworteten Ressourcen konnte mittels des T-Tests aufgrund der kleinen Testgruppe nicht festgestellt werden.

Insgesamt zeigt das Ergebnis der Benutzerstudie, dass die Qualität der Gruppierungen stark von den ausgewählten Ressourcen abhängt. Dabei gibt es keinen eindeutigen Hinweis darauf, wann die Bottom-Up oder die K-Means Variante in jedem Fall bessere Ergebnisse liefert. Hierfür wäre eine umfangreichere Studie durchzuführen, die die Qualität der erzeugten Ansichten im Hinblick auf die Art der Ressource untersucht.

4.4 Feedback von Testern

Nach der durchgeführten Evaluierung mittels des automatisierten Testprogramms gab es durch die Tester weitere Feedbacks, die hier kurz aufgenommen werden sollen.

Einige Tester bemängelten, dass die Gruppentitel oft zu verwirrend seien und den Inhalt der Gruppe manchmal unzureichend beschreiben würden. Dies führe dazu, dass Gruppen durchsucht werden, die die gesuchte Eigenschaft gar nicht enthalten. Präzisere Gruppentitel würden helfen den Suchaufwand zu minimieren. Auch sei es nicht verständlich, warum es Gruppen gibt, die den gleichen Titel haben.

Ein weiterer Kritikpunkt war, dass die gleiche Eigenschaft bei unterschiedlichen Ressourcen in unterschiedliche Gruppen einsortiert würde. Das mache die Suche noch schwieriger, da man nach bereits bekannten Gruppen suche. Die gesuchte Eigenschaft sei nicht mehr dort zu finden, sondern in einer anderen Gruppe. An dieser Stelle sei erwähnt, dass alle Tester jeweils drei unterschiedliche Ansichten mit unterschiedlicher Einsortierung gesehen haben. Es ist anzunehmen, dass hierdurch dem Tester die klare Abtrennung der unterschiedlichen Ansichten fehlt. Trotzdem kann dieses Phänomen auch bei der gleichen Art der Ansicht auftreten.

Bereits bei der Entwicklung und vor der Evaluierung wurde festgestellt, dass bei längeren Listen die Übersicht verloren geht. Dies wurde auch von den Testern bemängelt: Es sei langes scrollen notwendig, um die Liste zu überspringen und eine Anzeige aller Einträge in einer Liste sei nicht zielführend. Oft interessiere man sich nur für die wichtigsten Einträge einer größeren Liste. Eine Gesamtliste könne beispielsweise durch eine separate Anzeige oder durch Aufklappen dieser realisiert werden.

Positiv sei die separate Anzeige der Gruppen. Die Navigation durch Klicken auf die entsprechenden Gruppentitel sei sehr praktisch und intuitiv, und helfe bei der schnellen Navigation durch die Ressource.

Auch seien die Gruppentitel besonders dann vorteilhaft, wenn die Frage die Eigenschaft nicht direkt im Wortlaut enthält. Eine Suche nach der Eigenschaft würde sich ohne Gruppentitel als sehr mühsam herausstellen. Zur groben Orientierung helfe daher die Umschreibung in Form der Gruppentitel, um die Antwort auf die gestellte Frage zu finden.

4.5 Performance

Neben der Benutzerstudie wurde zudem die Performance der Implementierung getestet. Dazu wurden 100 zufällige Ressourcen von DBpedia ausgewählt. Gemessen wurde die benötigte Dauer für die Bildung der Gruppen, sowohl mit dem K-Means als auch mit dem Bottom-Up Clustering Algorithmus. Die Dauer für den Onlineabruf der Ressource wurde bewusst aus der Messung herausgenommen, da hier zu viele Parameter (Anbindung an das Internet, Erreichbarkeit von DBpedia, Auslastung des Netzes und viele weitere) die Dauer beeinflussen können.

Die Messung wurde auf einem Windows 7 Rechner durchgeführt, der mit einem Intel Core 2 Duo 2,4 GHz und 4 GB Arbeitsspeicher ausgestattet ist. Durchgeführt wurde die Messung mittels des Entwicklerservers der Springsource Entwicklungsumgebung des Grails Frameworks. Ein Deployment für den Produktivbetrieb würde auf einem Java Server wie Tomcat erfolgen. Dort ist aufgrund diverser Optimierungen mit einer höheren Performance zu rechnen.

Da die gemessenen Testdaten zufällig ausgewählt wurden, ist deren jeweiliger Umfang ebenfalls zufällig verteilt. Es ist aber zu erkennen, dass die meisten Ressourcen in DBpedia etwa von der Größenordnung 50 bis 200 Eigenschaften sind. Einzelne Ressourcen können aber unter Umständen mehrere tausend Eigenschaften groß sein. Die nachfolgende Abbildung 11 zeigt die gemessenen Werte der ausgewählten Ressourcen.

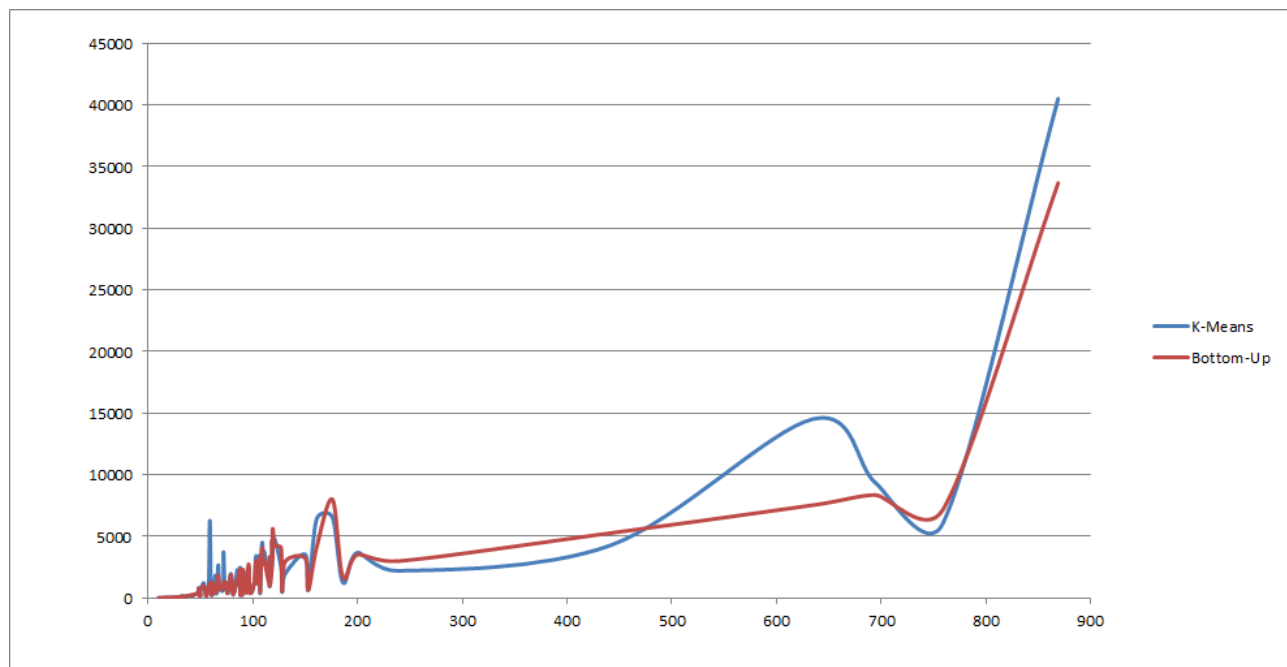


Abbildung 11: Gemessene Zeit für Gruppierungen in Millisekunden.

Man sieht recht deutlich, dass kein großer Unterschied zwischen den beiden Clustering Algorithmen festzustellen ist. Ebenso sieht man, dass bis etwa 450 Eigenschaften pro Ressource die Zeit zur Bildung von

Gruppen unter 5 Sekunden liegt. Erst ab 450 Eigenschaften wächst die benötigte Erzeugungszeit stärker an. Für die Ressource mit den meisten Eigenschaften (869) benötigten beide Clustering Algorithmen zwischen 30 und 40 Sekunden. Dies dürfte in erster Linie daran liegen, dass der für die Java Runtime bereitgestellte Speicher voll ist und die virtuelle Maschine Daten auslagern muss. Insgesamt wurden 94 der getesteten Ressourcen in weniger als 5 Sekunden angezeigt. Lediglich bei 6 Ressourcen muss der Nutzer länger als 5 Sekunden auf die Anzeige warten, unter Umständen sogar bis zu 40 Sekunden bei der größten Ressource. Mit einer langen Wartezeit für den Nutzer ist aber dennoch in den wenigsten Fällen zu rechnen.

5 Ausblick

Die in dieser Arbeit untersuchten Ansätze sind erst der erste Schritt für semantische Gruppierungen für Linked Open Data. Sie zeigen aber bereits, dass eine Gruppierung unter bestimmten Umständen einen Vorteil gegenüber der bisherigen Darstellung besitzt. Insbesondere die Qualität der Gruppierungen lässt noch Raum nach oben offen.

Eine Verbesserung ließe sich im Hinblick auf den Clustering Algorithmus umsetzen. So zeigte der CBC Algorithmus [Pantel, 2003] in einer anderen Arbeit, dass dieser semantische Ähnlichkeiten von Wörtern mittels WordNet mit höherer Präzision zuordnen kann. Eine um etwa 20 % höhere *Precision* [Pantel and Lin, 2002] im Gegensatz zum K-Means Clustering konnte bei den getesteten Worten festgestellt werden. Dies würde auch insgesamt zu einer präziseren Gruppenbildung führen. Der Clustering Algorithmus ist ein zentraler Bestandteil der Gruppierung von sinngemäß zusammengehörenden RDF-Tripel. Durch Einsatz anderer Algorithmen lässt sich die Gruppierung drastisch verändern und auch verbessern. Aber auch an den hier vorgestellten Algorithmen können Änderungen vorgenommen werden, um die Ergebnisse zu optimieren. Denkbar wäre beispielsweise eine Anpassung des Wertes k (bei K-Means) oder der Abschnittshöhe des Baums (bei Bottom-Up) abhängig von der Größe und Umfang der Daten. Auch könnte durch eine zusätzliche Nachbearbeitung der erzeugten Gruppen (beispielsweise das Zusammenlegen von gleichnamigen Gruppen) eine deutliche Verbesserung der Anzeige ermöglicht werden. Jeder Clustering Algorithmus ist aber auch auf die Korrektheit der Distanzfunktion angewiesen, um überhaupt erst strukturelle Zusammenhänge zu erkennen.

Die Distanzfunktion ist eine zentrale Funktion, die durch Optimierung und Anpassungen bessere Ergebnisse liefern kann. So existieren verschiedene Distanzen innerhalb von WordNet, die unterschiedliche Ergebnisse liefern [Budanitsky and Hirst, 2001]. Ebenfalls sollte über Alternativen zu WordNet nachgedacht werden. Der alternative Ansatz, mittels einer Suchmaschine Distanzen zwischen Wörtern zu berechnen, zeigte bereits in ersten Tests gute Ergebnisse. Eine Kombination von WordNet und Google ist eine denkbare Möglichkeit, um die Distanzberechnung zu verbessern. So erlaubt eine Kombination die Abschwächung von Fehlern, falls ein Ansatz abweichende Ergebnisse liefert. Eine weitere Möglichkeit ist die Nutzung von Wikipedia, um Sinnzusammenhänge zu erkennen. Mit *WikiRelate!* [Strube and Ponzetto, 2006] wurde eine Lösungsmöglichkeit vorgestellt, die vielversprechende Ergebnisse liefert. So fehlen in WordNet benannte Instanzen, Wikipedia kennt jedoch eine Vielzahl benannten Instanzen und spezielle Konzepte. Mittels interner Hyperlinks ließen sich Zusammenhänge erkennen und deren Ähnlichkeit feststellen. Dieser Ansatz dürfte insbesondere dann helfen, wenn es sich um Fachbegriffe oder ein fachspezifisches Vokabular handelt, das nicht in WordNet vorhanden ist.

Des Weiteren sollte eine Funktion die hohe Anzahl an gleichen Eigenschaften auf die wichtigsten reduzieren. Denn große Listen machen die Ansicht unübersichtlich und der Endbenutzer muss unter Umständen sehr lange scrollen, bis er die gewünschte Eigenschaft gefunden hat. Die Popularität von Ressourcen könnte dazu verwendet werden, um wichtige von weniger wichtigen Ressourcen zu unterscheiden. Hiermit ließe sich die Anzahl der angezeigten Einträge von langen Listen verkürzen. Als Maß für die Popularität könnte man die Anzahl der ein- und ausgehenden Kanten von Ressourcen nutzen. Zu wichtigen Daten liegen in der Regel mehr Informationen vor, als zu unwichtigen.

Neben der Gruppierung der Eigenschaften der Gruppen, wäre eine Sortierung der Gruppen nach Priorität eine weitere Optimierungsmöglichkeit. Wie in der Evaluierung festgestellt wurde, ist die Reihenfolge der Gruppen ein wichtiger Faktor, wie schnell bestimmte Eigenschaften gefunden werden können. Auch hier könnte man mittels der Anzahl an Verbindungen zwischen Ressourcen arbeiten. Ähnlich dem Page-Rank [Page et al., 1999] von Google ließe sich damit eine Maßzahl für die Wichtigkeit von Prädikaten angeben und dementsprechend eine Anordnung der Gruppen ermöglichen. Alternativ könnte der Typ der Ressource dazu verwendet werden, um die Relevanz von Ressourcen zu bestimmen. Dahingehend

wäre auch zu überlegen, ob man den Typ von Ressourcen separat und unabhängig von der Gruppenansicht anzeigt. Denkbar wäre eine Stichwortliste, die die aktuelle Ressource beschreibt. Damit erhält der Benutzer eine kurze prägnante Beschreibung der angezeigten Ressource.

Alle in dieser Arbeit vorgestellten Ansätze sind bislang nur für die englische Sprache und DBpedia umgesetzt worden. Sie sind jedoch so allgemein gehalten, dass ein Ausbau auf weitere Sprachen und weitere Datenquellen möglich ist. Der nächste Schritt ist also, für weitere Sprachen entsprechende semantische Wörterbücher zu nutzen und diese einzubauen. Diese müssen ähnlich dem WordNet die Möglichkeit zur Berechnung von Distanzen zwischen Wörtern ermöglichen. Eine Erweiterung um andere Datenquellen ist dagegen fast ohne Änderung der Implementierung möglich.

6 Fazit

Diese Arbeit hat gezeigt, dass man Gruppierungen von bereits existierenden Datensätzen in Linked Open Data vollständig automatisiert erzeugen kann. Dazu mussten weder manuell per Hand Informationen hinzugefügt werden, noch mussten Datensätze verändert werden. Mit der Nutzung des englischen Computerwörterbuchs WordNet in Verbindung mit einem Clusteringalgorithmus ist die Basis für das Finden von Gruppen gelegt worden. Die Qualität der erzeugten Gruppen ist in vielen Fällen überzeugend und zeigt das Potential der vorgestellten Ansätze. Ebenso ist die Erzeugung der Gruppen in recht kurzer Rechenzeit möglich. Dies zeigt, dass die vorgestellten Methoden auch in der Praxis einsetzbar sind. Beide Clusteringalgorithmen lassen sich gleich gut für das Finden von Gruppen für semantisch ähnliche Eigenschaften nutzen. Sowohl bei der Qualität der Gruppen, als auch bei der Performance konnten beide Ansätze überzeugen.

Das Austauschen des Wörterbuchs WordNet durch eine Suchmaschine hat sich als interessante Alternative herausgestellt. Sie ist allerdings aufgrund der beschränkten Schnittstellen der Suchmaschinenanbieter in der Praxis kaum einsetzbar. Die lange Wartezeit auf das Ergebnis und die enorme Anzahl der zu stellenden Anfragen an die Schnittstelle, machen den Ansatz weniger attraktiv. Dennoch konnte sich in den wenigen durchgeführten Tests die Qualität der erzeugten Gruppierungen sehen lassen. Auch wenn der Ansatz in dieser Arbeit aufgrund der starken Beschränkung der Schnittstellen nicht weiter verfolgt wurde, ist dieser dennoch nicht vollständig zu verwerfen.

Die durchgeführte Benutzerstudie hat gezeigt, dass eine simple Ansicht der Daten im Allgemeinen schneller zur Beantwortung von Fragen führt, als eine automatisch generierte Gruppenansicht. Zu vermuten ist, dass sich der Nutzer bei jeder Ansicht zunächst einen Überblick über die generierten Gruppen verschaffen muss, um schließlich die zugehörige Gruppe für die gestellte Frage zu finden. Diese Orientierungsphase entfällt im Wesentlichen bei einer alphabetischen Ansicht, da der Benutzer nur zur entsprechenden Eigenschaft, die in der Frage erwähnt wird, wechseln muss. Dementsprechend lassen sich Fragen, die eine bestimmte Eigenschaft direkt abfragen, schneller beantworten, da die vorherige Suche der richtigen Gruppe entfällt. Dabei sollte nicht vernachlässigt werden, dass eine alphabetische Liste eine vertrautere Ansicht ist, da Menschen solche häufig im Privat- und Geschäftsleben vorfinden (z.B. Telefonbuch, Kundendatei usw.).

Auf der anderen Seite konnte die Benutzerstudie aber auch belegen, dass für bestimmte Ressourcenarten die gruppierte Ansicht bessere Ergebnisse liefert. So ließen sich Fragen zur Kategorie Länder schneller mit einer gruppierten Ansicht beantworten, als mit einer konventionellen sortierten Liste. Die Qualität der Gruppierung hängt im Wesentlichen auch von der Datenqualität der betrachteten Ressource und insbesondere von der Qualität der Prädikatenbeschreibung ab. Abkürzungen oder zu knappe Beschreibungen führen dazu, dass Eigenschaften falsch zugeordnet werden, was wiederum für eine schlechtere Anzeige führt. Mittels Abkürzungslexikon, Stemmer, Nominalisierung und Rechtschreibkorrektur lassen sich zusätzliche Informationen gewinnen, um die semantische Bedeutung zu präzisieren. Damit lassen sich allerdings nur wenige Zuordnungen verbessern.

Die Benutzerstudie hat nicht eindeutig belegt, dass durch maschinen-generierte Gruppen schnellere und qualitativ bessere Antworten auf Fragen zu erreichen sind. Sie zeigt jedoch deren grundsätzliche Realisierbarkeit. Zwar konnte der Anwendungsfall „Quiz“ nicht direkt überzeugen, vorstellbar ist jedoch der Einsatz im Bereich der allgemeinen Informationsdarstellung für den Menschen. Die heutigen Linked Open Data Datenrepräsentationen sind im Hinblick auf ihre Benutzerfreundlichkeit verbesserungsfähig. Der Einsatz, der in dieser Arbeit vorgestellten Ansätze, könnte deshalb dazu beitragen, dass auch Nicht-informatiker die Daten von Linked Open Data nutzen und verstehen können. Gerade weil Linked Open Data viele Millionen Informationen semantisch greifbar verfügbar macht, ist es kaum realisierbar, diese Daten manuell für den Benutzer aufzubereiten. Deswegen müssen Algorithmen und Automatismen, wie hier vorgestellt, gefunden werden, um die Daten für den Menschen lesbar zu machen.

Literatur

- [Wol, 2012] (2012). *About Wolfram|Alpha: Making the World's Knowledge Computable*. <http://www.wolframalpha.com/about.html>.
- [Auer et al., 2007] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). *DBpedia: A Nucleus for a Web of Open Data*.
- [Berners-Lee, 2005a] Berners-Lee, T. (2005a). *Notation 3 Logic*. <http://www.w3.org/DesignIssues/Notation3>.
- [Berners-Lee, 2005b] Berners-Lee, T. (2005b). *Uniform Resource Identifier (URI): Generic Syntax*. <http://tools.ietf.org/html/rfc3986>.
- [Berners-Lee, 2006] Berners-Lee, T. (2006). *Linked Data*. <http://www.w3.org/DesignIssues/LinkedData.html>.
- [Berners-Lee et al., 2006] Berners-Lee, T., Chen, Y., Chilton, L., Connolly, D., Dhanaraj, R., Hollenbach, J., Lerer, A., , and Sheets, D. (2006). *Tabulator: Exploring and Analyzing linked data on the Semantic Web*. Cambridge, MA, USA.
- [Bizer and Gauß, 2007] Bizer, C. and Gauß, T. (2007). *Disco - Hyperdata Browser: A simple browser for navigating the Semantic Web*. <http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/>.
- [Budanitsky and Hirst, 2001] Budanitsky, A. and Hirst, G. (2001). *Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures*. University of Toronto.
- [Cilibrasi and Vitányi, 2004] Cilibrasi, R. and Vitányi, P. M. B. (2004). The google similarity distance. *CoRR*, abs/cs/0412098.
- [Cyganiak and Jentzsch, 2011] Cyganiak, R. and Jentzsch, A. (2011). *Linking Open Data cloud diagram*. <http://lod-cloud.net/>.
- [Fellbaum and Miller, 1998] Fellbaum, C. and Miller, G. (1998). *WordNet - An Electronic Lexical Database*. MIT Press, Cambridge.
- [Graham Klyne, 2004] Graham Klyne, J. J. C. (2004). *Resource Description Framework (RDF): Concepts and Abstract Syntax*. World Wide Web Consortium (W3C), <http://www.w3.org/TR/rdf-concepts/>.
- [Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning - Data Mining, Inference, and Prediction, Second Edition*. Springer, Berlin, Heidelberg, 2nd ed. 2009. corr. 3rd printing 5th printing. edition.
- [Hitzler et al., 2007] Hitzler, P., Krötzsch, M., Rudolph, S., and Sure, Y. (2007). *Semantic Web: Grundlagen (eXamen.press) (German Edition)*. Springer.
- [Jentzsch, 2009] Jentzsch, A. (2009). *DBpedia – Extracting structured data from Wikipedia*. Presentation at Semantic Web.
- [Klyne and Carroll, 2004] Klyne, G. and Carroll, J. J. (2004). Resource description framework (rdf): Concepts and abstract syntax. <http://www.w3.org/TR/rdf-concepts/>.
- [Miller, 1956] Miller, G. A. (1956). *The magical number seven, plus or minus two: Some limits on our capacity for processing information*. Harvard University. *Psychological Review*.
- [Nuzzolese et al., 2011] Nuzzolese, A. G., Gangemi, A., Presutti, V., and Ciancarini, P. (2011). *Encyclopedic Knowledge Patterns from Wikipedia Links*.

-
- [Page et al., 1999] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). *The PageRank Citation Ranking: Bringing Order to the Web*. Previous number = SIDL-WP-1999-0120.
- [Pantel and Lin, 2002] Pantel, P. and Lin, D. (2002). *Discovering Word Senses from Text*. University of Alberta.
- [Pantel, 2003] Pantel, P. A. (2003). *Clustering by Committee*. PhD thesis, Edmonton, Alberta.
- [Paulheim, 2011] Paulheim, H. (2011). Einführung in Semantic Web Vorlesung. Foliensatz.
- [Seeliger and Paulheim, 2012] Seeliger, A. and Paulheim, H. (2012). A Semantic Browser for Linked Open Data. In *Semantic Web Challenge*.
- [Singhal, 2012] Singhal, A. (2012). *Introducing the Knowledge Graph: things, not strings*. <http://googleblog.blogspot.de/2012/05/introducing-knowledge-graph-things-not.html>.
- [Strube and Ponzetto, 2006] Strube, M. and Ponzetto, S. P. (2006). *WikiRelate! Computing Semantic Relatedness Using Wikipedia*. EML Research gGmbH.
- [Witten et al., 2005] Witten, I. H., Frank, E., and Hall, M. A. (2005). *Data Mining - Practical Machine Learning Tools and Techniques*. Elsevier, Amsterdam.

Abbildungsverzeichnis

1	Verlinkungen von und zu DBpedia. Quelle: [Cyganiak and Jentzsch, 2011].	5
2	DBpedia-Ansicht zum Eintrag Darmstadt.	7
3	aemoo Ansicht von der Ressource Deutschland.	8
4	Die Freebase-Ansicht zum Eintrag Darmstadt.	9
5	Aufbau der Implementierung des sematischen Browsers.	18
6	Webansicht des semantischen Browsers der Ressource <i>Darmstadt</i> . Zu sehen sind die erzeugten Gruppen, die Navigationsleiste (links) und die RDF-Tripel.	19
7	Evaluationsprogramm. Es zeigt die Ressource <i>Kentucky</i> in der <i>Bottom-Up</i> Ansicht.	23
8	Verteilung der Benutzerbewertungen nach Ansicht.	24
9	Durchschnittliche Anzahl an Gruppen, gruppiert nach Städten, Ländern und Filmen. . . .	26
10	Durchschnittliche Beantwortungsdauer, gruppiert nach Städten, Ländern und Filmen. . . .	26
11	Gemessene Zeit für Gruppierungen in Millisekunden.	27

Tabellenverzeichnis

1	Abhängigkeit der Gruppenanzahl und -größe von der gewählten Baumhöhe.	16
2	Gruppenanzahl und -größe bei Baseline	21
3	Gruppenanzahl und -größe bei K-Means	22
4	Gruppenanzahl und -größe bei Bottom-Up	22
5	Verteilung der Darstellungsformen bei der Evaluierung.	22
6	Bewertung der Ansichten durch die Tester.	24
7	Ergebnis der Benutzerstudie. Quelle: [Seeliger and Paulheim, 2012]	25

7 Anhang

Benutzerstudie: Verwendete DBpedia Ressourcen und dazu gestellte Fragen

Brownsville,_Kentucky In which time zone is Kentucky?	Amsterdam How large is the density of population in Amsterdam?
Restoration_and_Regeneration_(Switzerland) When started and ended the Swiss Confederation?	Sydney How many schools does have Sydney?
Later_Qin Was Later Qin a Monarchy?	The_Shooter_(1995_film) When is The Shooter released and who produced it?
Curlew,_Iowa What is the area and postal code of Curlew, Iowa?	Nixon,_Texas What is the area and postal code of Nixon, Texas?
Cameroon What is the capital of Cameroon?	Republic_of_Zamboanga Where was the Republic of Zamboanga and what language was spoken?
Garfield_Gets_Real Who is the producer and director of Garfield Gets Real?	Fiji What is the biggest city of Fiji? Is it the capital?
Parkersburg,_Iowa How many people live in Parkersburg, Iowa?	Police_(1916_film) Who wrote the Police (film)?
Thomond What country is Thomond today?	Minority_Report_(film) How large was the budget of Minority Report?
Netherlands_New_Guinea When was Netherlands New Guinea founded?	Corona,_California Corona, California is also called 'The ...'
Herbie_Goes_to_Monte_Carlo What is the duration in minutes of Herbie Goes to Monte Carlo?	Warfield,_Kentucky What is the area and postal code of Warfield, Kentucky?
Ames,_Iowa Name one person who was born in Ames, Iowa?	Bad_Company_(2002_film) Who wrote the story of Bad Company and what company distributed it?
First_Hellenic_Republic What was the name of the First Hellenic Republic money?	10,000_BC_(film) Who made the music of 10.000 BC?
Harvard,_Nebraska Who is the mayor of Harvard, Nebraska?	Faroe_Islands What is the currency name of the Faroe Islands?
Kingdom_of_Galicia What is the capital and the national language of Kingdom of Galicia?	Shadows_(1959_film) Who produced the Shadows (film) and in which country?
Six_Days_Seven_Nights Who made the music of the Six Days, Seven Nights?	Casino_Royale_(1967_film) Was Joanna Pettet an actor starring in Casino Royale (1967)?