Aufbau von Benutzerprofilen für personalisierte Systeme mit Hilfe von Semantic Web Technologien

Bianca Gotthart

MASTERARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

Interactive Media

in Hagenberg

im September 2013

Erklärung

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Hagenberg, am 29. September 2013

Bianca Gotthart

Inhaltsverzeichnis

Kurzfassung

Filter- und Personalisierungssysteme passen Information an die Bedürfnisse des Benutzers an, um die Informationsbeschaffung für diesen zu erleichtern. Die Problematik dabei ist, wie zu den Interessen des Benutzers gelangt wird, ohne auf regelmäßige und explizite Handlungen des Benutzers angewiesen zu sein. Benutzerprofile sind oftmals entweder zu eingeschränkt oder spezifisch angelegt, was für Systeme wenig Spielraum für die Personalisierung bietet.

Im Zuge dieser Arbeit wurde eine Methodik konzipiert, wie Benutzerdaten mit der Hilfe einer Google Chrome Browser Erweiterung automatisch ermittelt werden können. Dafür wurden Semantic APIs verwendet, welche Inhalte von besuchten Webseiten analysieren und eine semantische Textextraktion durchführen. Die extrahierten Daten wurden anschließend mit der Wissensbasis von Wikipedia angereichert und in einem Benutzerprofil gespeichert. Semantic Web Technologien unterstützen bei dem Aufbau des Profils, um die Information austauschbar aufzubereiten und im Web verfügbar zu machen. Der Grund für die weitere Anreicherung ist, dass somit das Profil mit mehr Wissen ausgestattet wird und die einzelnen Interessen mit weiteren Begriffen in Verbindung gebracht werden können.

Mit Hilfe des Benutzerprofils wäre es beispielsweise für einen personalisierten Newsfeed möglich, das Benutzerprofil als Basis für den Personalisierungsprozess zu verwenden, um Information sowohl nach allgemeinen, als auch spezifischen Interessen zu filtern.

Der konzipierte Ansatz wurde evaluiert und zeigt, dass eine Anreicherung mit Wikipedia Kategorien das Benutzerprofil mit Begriffen erweitert, welche die Interesse des Benutzers widerspiegeln und verwandte Themengebiete liefern.

Abstract

Personalisation systems adjust the information flow based on the needs of the user in order to more easily obtain information. The problem here is that knowledge about the user needs to be gathered without having to rely on regular and explicit user feedback. Furthermore, user profiles are often either too limited or specifically created for systems which offers little scope for personalization.

In the course of this thesis, a methodology is designed to automatically identify user data with the help of a Google Chrome browser extension. Semantic APIs are used to extract important semantic items from the content of visited websites. In addition, the extracted data is enriched with concepts from the knowledge base of Wikipedia and stored in an user profile. Semantic Web technologies help in constructing the profile to be exchangeable and make the data available on the web. The reason for the additional enrichment is that the profile is equipped with more knowledge and allowing individual interests to be associated with other terms of the Wikipedia database and be linked in a network of topics within the profile.

It would be possible to use the resulting user profile and extracted interests for personalized news feed generation in order to filter information according to both general and specific interests.

The designed approach was evaluated and is showed that because of the additional enrichment with Wikipedia categories the user profile is being expanded with terms that reflect the user's interests and related areas.

Kapitel 1

Einleitung

1.1 Problemstellung und Motivation

Im letzten Jahrzehnt hat sich die Anzahl der Daten im Web stark vermehrt. Die große Menge an Information führt allerdings dazu, dass die Informationen sehr verteilt sind und eine Informationsüberflutung herrscht. Auch Suchmaschinen geben mehr als 1500 Ergebnisse pro Anfrage zurück, welches die Informationsbeschaffung nicht unbedingt erleichtert. Deshalb ist es in Zukunft von besonderer Bedeutung, dass es Filter- und Suchsysteme gibt, welche an den Bedürfnissen der einzelnen Benutzer angepasst werden, um personalisiert Informationen liefern zu können. Dafür muss ein System Wissen über den Benutzer erlernen und sich laufend anpassen, sodass der Personalisierungsprozess verbessert werden kann. Viele Wissensbibliotheken haben im Web eine besondere Bedeutung gewonnen, wie zum Beispiel Wikipedia, eine Enzyklopädie, welche eine große Menge an unterschiedlichen Themengebieten abdeckt. Wikipedia ist auch für Google eine vertrauenswürdige Quelle und wird in sehr vielen Suchergebnissen an oberster Stelle angeboten. Die Daten werden in strukturierter Form öffentlich zur Verfügung gestellt, womit unterschiedliche Anwendungen dies als Grundlage für semantische Analysen oder zur Informationsanreicherung verwenden [6].

Mit der Idee des Semantic Webs von Berners-Lee et al. [3] sollen Informationen den Bedürfnissen des Benutzers angepasst werden. Dieser Gedanke setzt voraus, dass Maschinen Informationen verstehen und zueinander in Verbindung setzen können. Dafür müssen die Daten, sowohl vom Benutzer als auch von anderen Ressourcen in strukturierter Form vorhanden sein. Personalisierung spielt für eine bessere Nutzung des Webs eine immer bedeutsamere Rolle. Viele Anwendungen beinhalten Benutzerprofile, um Wissen von Benutzern zu sammeln und als Wissensbasis für den Personalisierungsprozess zu verwenden.

Somit stellt sich die Frage, in welcher Form und mit welchen semantischen Hilfsmitteln ein Benutzerprofil aufgebaut werden kann, um die Interessen ab-

zubilden, sodass Informationssysteme dies als Basis für die Personalisierung verwenden können.

1.2 Ziel und Lösungsansatz

Die Zielsetzung in dieser Arbeit ist, ein Profil mit den Interessen des Benutzers automatisiert zu erstellen, um dies als Grundlage für die Personalisierung von Informationen zu verwenden. Mit diesem Benutzerprofil soll ermöglicht werden, dass Informationen individueller aufbereitet werden, um die Zufriedenheit des Benutzers zu steigern. Da der Aufbau einer Wissensbasis über den Benutzer in Form von Benutzerprofilen ein Grundstein für die Personalisierung im Web darstellt, beschäftigt sich diese Arbeit mit den dafür notwendigen Schritten. Dafür müssen Informationen aus persönlichen Quellen extrahiert bzw. erlernt und gespeichert werden. Als Quelle für diese Interessensermittlung werden die besuchten Webseiten vom Benutzer verwendet. Mit der Hilfe von bereits existierenden Wissensdatenbanken und Semantic Web Technologien sollen die Interessen automatisiert ermittelt werden. Wikipedia als Wissensdatenbank soll eine Verbindung zu weiteren Informationen herstellen, um das Profil zu erweitern. Wikipedia besitzt ein sehr breites Informationsspektrum mit Beziehungen zu weiteren Begriffen und bietet mit DBpedia eine Schnittstelle für den Zugriff der Daten in öffentlicher strukturierter Form. Außerdem soll gezeigt werden, welchen Ausmaß es hat, wenn nicht nur einzelne direkt extrahierte Wörter als Interessen im Profil gespeichert werden, sondern auch verwandte, automatisch extrahierte verbundene Konzepte aus Wikipedia mitbestimmend sind.

1.3 Inhaltlicher Aufbau

Die Arbeit ist in sieben Kapitel unterteilt. Das nachfolgende Kapitel behandelt Grundlagen sowohl aus dem Bereich Semantic Web, als auch zum Thema Personalisierung im Web und stellt eine Einführung in das Themengebiet dar. Bei der Personalisierung wird besonders auf die Methoden für die Generierung von Interessen des Benutzers eingegangen, aber auch auf die unterschiedlichen Arten von Benutzerprofilen. Im Fokus des dritten Kapitels stehen verschiedene Ansätze der Personalisierung im Web und der Aufbau von Benutzerprofilen in Verbindung mit dem Themengebiet Semantic Web. Das vierte und fünfte Kapitel fokusieren sich auf den eigenen Ansatz in Form eines Entwurfskonzepts mit anschließenden Details der Implementierung eines Benutzerprofils, welcher im sechsten Kapitel anhand eines Testprofils evaluiert wird. Den Abschluss der Arbeit bildet ein Resümee, worin sich eine Zusammenfassung, ein Ausblick auf mögliche Erweiterungen und ein Fazit befinden.

Kapitel 2

Grundlagen

Im nachfolgenden Kapitel werden die Grundlagen, welche für diese Arbeit relevant sind, behandelt. Diese deckt sowohl Themenbereiche des Semantic Webs, als auch Personalisierung im Web ab. Der Fokus liegt auf die Extraktion von Benutzerdaten und auf der Speicherung von Benutzerprofilen.

2.1 Semantic Web

2.1.1 Vision des Semantic Webs

Im Word Wide Web kursiert eine große Menge an Information. Wenn nach einer Information gesucht wird, liefern Suchmaschinen viele Verlinkungen auf den unterschiedlichen Webseiten zurück. Unter diesen Ergebnissen befinden sich allerdings viele, die nicht unmittelbar relevant sind, da Suchkriterien oftmals mit vielen Details verknüpft sind. Diese auszusortieren und die tatsächlich relevanten Links zu finden, kann oftmals zu einer Hürde werden.

Die wichtigste Eigenschaft des Word Wide Webs ist dessen Allgemeinheit und Vielseitigkeit, sodass mit Hilfe von Technologien wie Hypertext-Verlinkungen Informationen miteinander verknüpft werden können. Eine Bedeutung zu dieser Verknüpfung herzustellen oder zu sagen, in welcher Art diese Informationen zusammengehören, ist für einen Menschen möglich. Damit Maschinen dies automatisch interpretieren können, müssen die Daten in einer besonderen Form semantisch annotiert werden.

Tim Berners-Lee wollte dieses Problem mit seinem Konzept des *Semantic Webs* 2001 lösen. Der Zugriff auf Informationen soll für die Benutzer vereinfacht werden, indem semantische Hintergründe bei der Speicherung und Abfrage von Informationen einfließen, diese allerdings auch mit Hilfe von automatisierten Prozessen von Maschinen interpretierbar sind [26].

Um dies zu erreichen, müssen die Informationen in strukturierter Form angeboten werden. Mit Hilfe von Schlussfolgerungsregeln und allgemein definierten Konzepten müssen die Daten aufbereitet werden, um automatisiert eine Bedeutung zuordnen zu können und somit die angeforderte Information

ausfindig zu machen [3].

Im folgenden Abschnitt wird ein kurzer Überblick über die Technologien des Semantic Webs gegeben.

2.1.2 Standardisierte Komponenten des Semantic Webs

Inhalte im Web werden für Menschen aufbereitet, um sie für diese konsumierbar zu machen, um die Daten beschreiben zu können. Diese Metadaten beschreiben somit die Semantik der Daten, wodurch ermöglicht wird, dass Ressourcen auffindbar und Beziehung zueinander für Maschenen verständlich aufbereitet werden [26]. Für die Auszeichnung der Daten ist eine Standardisierung notwendig.

XML (eXtensible Markup Language)

XML ist eine Empfehlung des *World Wide Web Consortium*[1] (W3C) für den Austausch von Daten und dient als Basis für dessen Annotation. Grundsätzlich ist es eine Markupsprache, die Daten beinhaltet, die wiederum Daten näher beschreiben. Diese werden auch als Metadaten bezeichnet. Die Informationen werden innerhalb einer grundlegenden logischen Struktur gegliedert, sodass Maschinen diese interpretieren können. Allerdings fehlt diesem Standard die Möglichkeit, den Daten auch eine Bedeutung zuzuordnen [19].

RDF (Resource Description Framework

RDF ist ebenfalls eine Spezifikation des W3C und ist eine allgemeine Methode für die Beschreibung von Ressourcen im Web. Das Ziel von RDF ist, dass die Daten von unterschiedlichen Quellen in einheitlicher Art und Weise zugänglich gemacht werden, ohne Verlust deren Bedeutung und Beziehungen zu anderen [42].

Die Daten werden in sogenannten *Triples* eingeteilt und mit XML ausgezeichnet. Ein *Triple* beinhaltet ein Subjekt, ein Prädikat und ein Objekt. Die einzelnen Elemente sind mit *Universal Resource Identifier* (URI) eindeutig identifizierbar und werden verwendet, um Informationen mit Hilfe des Hypertext Transfer Protokol (HTTP) im Web ausfindig zu machen und diese zu verknüpfen. Das Subjekt, Prädikat und Objekt können unterschiedliche Ressourcen im Web repräsentieren, wie zB eine URL einer Webseite, ein Dokument, eine Person. Im Web sollen Ressourcen von unterschiedlichsten Quellen miteinander vernetzt werden [3]. Nähere Informationen bezüglich XML und RDF sind deren Spezifikation zu entnehmen [43].

Die im RDF-Format ausgezeichneten Daten werden in sogenannten RDF Stores oder Triple Stores gespeichert und zur Wiederverwendung und Verknüpfung mit anderen Datenquellen online zur Verfügung gestellt [19].

[1] W3C: http://www.w3.org/

Ontologie

Ein weiteres Konzept von Semantic Web sind Ontologien. Diese ermöglichen es, Wissen mit Hilfe von Konzepten zu kategorisieren und in eine dafür passenden Domänen abzubilden. Mit der *Web Ontology Language* (OWL) können Entitäten in Klassen oder Subklassen abgebildet, Eigenschaften und Beziehungen zugeordnet und Operationen ausgeführt werden [41, 26].

SPARQL

Um Daten aus einer RDF-Struktur abfragen zu können, wird die Abfragesprache *SPARQL Protocol and RDF Query Language* (SPARQL) verwendet. Damit können Daten aus einer Triple-Datenbank abgefragt und manipuliert werden [19]. Nähere Informationen bezüglich SPARQL sind in dieser Spezifikation [45] zu entnehmen.

2.1.3 Linked Data

Als *Linked Data*[2] werden Daten in RDF-Form bezeichnet, die im Web frei verfügbar und mit anderen Daten aus unterschiedlichen Datenquellen verbunden sind. Tim Berners-Lee hat vier Prinzipien definiert [37]:

- URIs sollen für die Namen von Dingen verwendet werden.
- URIs sollen als HTTP URIs ausgezeichnet werden, damit diese nachgeschlagen werden können.
- Die hinten den URIs verknüpften Informationen sollen mit den Standards RDF und SPARQL angeboten werden.
- Die weiteren Ressourcen sollen ebenfalls mit URIs ausgezeichnet sein, um auch diese nachschlagen zu können.

Linked Open Data Project

Das größte Projekt, welches die Linked Data Prinzipien anwendet, ist das Linked Open Data Project (LOD)[3]. Es ist ein vom W3C 2007 gegründetes Projekt mit dem Ziel, frei verfügbare Datensätze im Web als RDF-Daten zu kodieren und zu veröffentlichen.

Es gibt eine große Anzahl an Daten, die im Web in dieser Form dargestellt werden, wodurch bereits viele Domänen abgedeckt sind: geographische Informationen, Personen, Unternehmen, Filme, Musik, Bücher, wissenschaftliche Publikationen, Regierungsdaten, uvm. [4, 38].

Abbildung 2.1 zeigt eine Übersicht über die größten RDF-Triple Datenquellen im Web. Jeder Knoten zeigt einen eindeutigen Datensatz an, die als Linked Data veröffentlicht wurden. Die Kanten (Pfeile) symbolisieren die

[2] http://linkeddata.org/
[3] http://linkeddata.org/

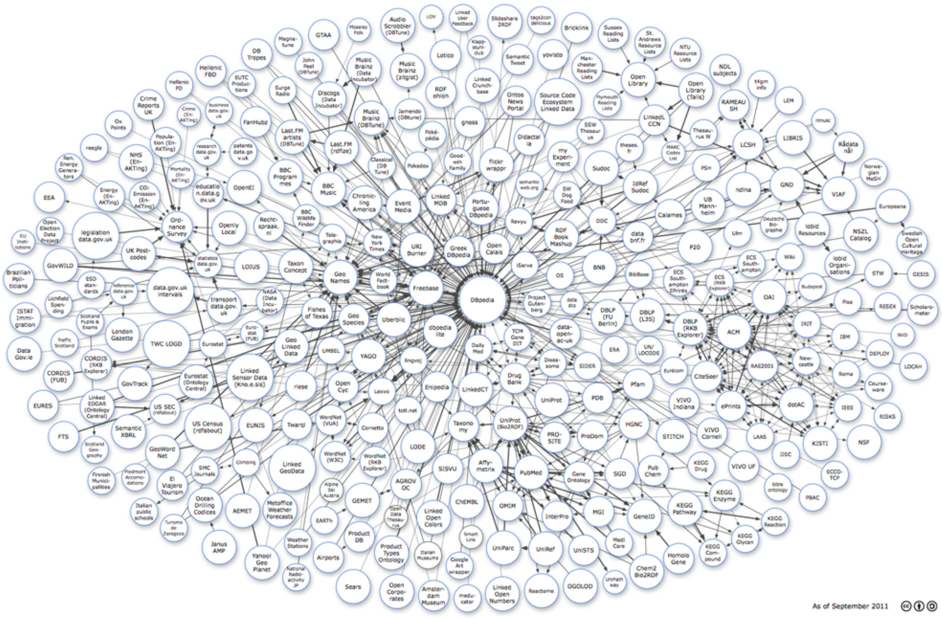

Abbildung 2.1: Die Linked Open Data Cloud mit dem Stand von September 2011 [39].

Verknüpfung der Datenquelle. Je dicker der Pfeil, desto größer ist die Anzahl an Verknüpfungen [5]. Im September 2011 wurden die Größe der LOD auf über 31 Milliarden RDF-Triples geschätzt[4]. Die größten Datenquellen innerhalb der *Linked Open Data* sind die folgenden: BBC, MusicBrainz, Thomson Reuters, DBpedia uvm. [37].

DBpedia

In dieser Arbeit steht ein besonderes Konzept der LOD im Vordergrund, welches in diesem Abschnitt näher beschrieben wird. DBpedia[5] ist eine Crowdsourcing Gemeinschaft und ist im *LOD* Projekt integriert. Es bildet die gesamte Wikipedia Online-Enzyklopädie als Ontologie ab und stellt die Daten in RDF Form zur Verfügung [34]. Diese Ontologie beinhaltet 103 Millionen RDF-Triple, die in eigenen Projekten integriert werden können. Darüber hinaus sind die Daten auch mit anderen Datenquellen verknüpft, wodurch insgesammt über 2 Milliarden RDF-Triple miteinander in Verbindung stehen [1].

Für Semantic Web Anwendungen ist es wichtig, dass auf eine große Wissensbasis zurückgegriffen werden kann, um für Analysen eine große Menge an unterschiedlichen Informationen anzubieten. DBpedia bietet die Daten in

[4] http://lod-cloud.net/state/
[5] http://dbpedia.org/

verschiedenen Domänen und Sprachen an, die regelmäßig aktualisiert und verifiziert werden [6].

Die DBpedia Datensätze beinhalten folgende Informationen[6]:

- Artikel
- Kurzfassung des Artikels
- Sprachen und Übersetzungen
- Infobox Inhalte
- externe Links zu Webseiten
- Kategorie des Artikels
- Informationen über die Kategorie selbst
- Personen in Form des FOAF-Konzepts
- Verlinkungen innerhalb der Seite/des Artikels zu DBpedia Ressourcen
- RDF Verlinkungen

Jede Ressource innerhalb von DBpedia ist als URI in dieser Form `http://dbpedia.org/resource/`*`Name`* identifiziert und über HTTP verfügbar. *Name* symbolisiert den Namen der Ressource.

Open Directory Project

Das *Open Directory Project*[7] – auch genannt als *ODP* oder *DMOZ* – ist ein Open Source Projekt, dass über 4 Millionen Webseiten in 590.000 Kategorien beinhaltet. Die Datenstruktur ist als Baum aufgebaut, worin die Kategorien die Knoten, und die Webseiten die Blätter repräsentieren. Knoten können auf Grund von symbolischen Verlinkungen zu mehreren Elternknoten gehören. Dieses Projekt unterstützt mehrere Sprachen, wie zB Deutsch, Englisch, Japanisch, Spanisch uvm. [40].

WordNet

WordNet[8] ist eine große lexikalische Datenbank und beinhaltet englischen Nomen, Verben, Adjektive und Adverbien. Diese Begriffe sind in Mengen von Synonymen (den sogenannten *synsets*) gruppiert. Jedes einzelne drückt ein eindeutiges Konzept aus. Diese Synsets sind entweder über eine lexikalische Beziehung oder auf Grund von einer konzeptionell-semantischen Beziehung miteinander verbunden. Dieses Netzwerk an bedeutungsvollen verknüpften Konzepten unterstützt Prozesse der Computerlinguistik aber auch die Verarbeitung der natürlichen Sprache. Die Datenbank ist online frei verfügbar und kann heruntergeladen werde, um sie in eigenen Anwendungen zu integrieren [48].

[6] Mehr Informationen zu den DBpedia Datensätze: http://wiki.dbpedia.org/Datasets, zuletzt aufgerufen am 24.4.2013

[7] http://www.dmoz.org/

[8] http://wordnet.princeton.edu/

2.2 Web Personalisierung

Dieser Abschnitt gibt einen Überblick über die wichtigsten Aspekte der Personalisierung mit dem Fokus auf den Aufbau von Benutzerprofilen.

2.2.1 Gründe und Ziele der Personalisierung

Der Begriff Personalisierung im Web umfasst Methoden und Technologien, um Web-basierte Informationssystem an die Bedürfnisse und Interessen von Benutzern anzupassen. Im Web gibt es sehr viele Informationen, die für eine Person relevant sind. Um diese zu filtern und somit einfacher zugänglich zu machen, ist Personalisierung von Informationssystemen nötig. Personalisierungssysteme filtern die relevanten Informationen individuell und identifizieren jene Informationen, die sich mit den Eigenschaften und Interessen des Benutzers decken. Es gibt unterschiedliche Forschungsgebiete, die sich mit Personalisierung beschäftigen. Dazu gehören Bereiche wie Information Retrieval, Artificial Intelligence, Data Mining, uvm. [17].

2.2.2 Strategien von Personalisierung

Personalisierung kann nach unterschiedlichen Strategien angewendet werden, je nachdem, wie stark die Personalisierung in das System eingreift und wie viel an Wissen vom Benutzer vorhanden ist [27].

Memorization

Der Memorization-Strategie zur Folge werden Informationen über den Benutzer, wie zB Name oder Browsing History, im System gespeichert. Die Daten werden in keiner Weise verarbeitet. Sie dienen nur der Wiedererkennung des Benutzers. Anwendungsgebiete dieser Strategie sind zB Begrüßungstext und Bookmarking von bereits besuchten Seiten. Für die direkte Anrede des Benutzers wird dessen Name verlangt. Dies kann nur über direkte Eingabe vom Benutzer erfolgen. Der Benutzername kann entweder in einer Datenbank oder in einer Cookie-Datei gespeichert werden. Eine Kontrolle der Übereinstimmung des Benutzernamen mit dem aktiven Benutzer ist wichtig, um einen unbefugten Zugriff zu vermeiden.

Guidance

Bei der Guidance-Strategie soll dem Benutzer geholfen werden, schneller zu gesuchten und für den Benutzer relevanten Informationen zu gelangen wie zB mit dem Angebot von Link-Empfehlungen, die für den Benutzer von Interesse sein könnten. Dafür wird Wissen darüber benötigt, was der Benutzer auf einer Webseite gerade sucht. Wichtig dabei ist, dass die Identifikation

von verwandten Dokumenten erleichtert wird, indem Muster oder Zusammenhänge zu anderen Dokumenten betrachtet werden.

Customization

Diese Strategie verweist auf eine aktive Anpassung des Systems im Zuge von Veränderungen am Inhalt, an der Struktur oder am Layout, in Abhängigkeit des Wissens über den Benutzer. Dafür werden dessen Interessen und Eigenschaften benötigt, die zumeist mit Hilfe der Browsing History erlangt werden. Kategorisierung oder Clustering von Webseiten werden hierbei oft verwendet.

Task Performance Support:

Systeme, die mit Task Performance Support Personalisierung anwenden, führen Aktionen für den Benutzer automatisch durch, wie zB personalisierte Anpassung von zusätzlichen Suchparametern, um die Suche einzuschränken. Hierfür ist die Intention des Benutzers, warum eine gewisse Aktion durchgeführt wurde, von Bedeutung. Mit einer Analyse des Browsing Verhaltens kann die Intention des Benutzers herausgefunden werden, wobei die Kenntnis über den typischen Navigationspfad des Benutzers wichtig ist, um eine Aktion für diesen auszuführen.

2.2.3 Ansätze von personalisiertem Filtern

Um Personalisierung durchzuführen, werden unterschiedliche Methoden angewendet. Diese unterscheiden sich in der Art und Weise, welche Informationen personalisiert werden sollen [24]. hat die Möglichkeiten, wie personalisierte Inhalte gefiltert werden können, mit Hilfe von drei allgemeinen Ansätzen definiert. Diese Ansätze schließen sich nicht untereinander aus. Auch Kombinationen werden oft angewendet.

Rule-based Personalization System

Regelbasierte personalisierte Systeme benötigen entweder manuelle oder automatisch generierte Entscheidungsregeln, die verwendet werden, um Inhalte für einen Benutzer zu empfehlen. Viele Online-Shop Systeme basieren auf einem derartigen Ansatz. Die Webseitenbetreiber definieren spezielle Regeln, nach denen die Inhalte für den Benutzer gefiltert werden. Diese Regeln basieren auf den Eigenschaften des Benutzers, die meist auf demographischen, geographischen aber auch persönlichen Eigenschaften des Benutzers basieren. Diese Daten werden mit einem Formular zur Bildung eines statischen Profils ermittelt. Die Eingabe der eigenen Daten ist üblicherweise subjektiv, wodurch dies anfällig ist für Verzerrungen. Außerdem sind die Profile oft

statisch, wodurch das Profil veraltet sein kann, wenn es über einen längeren Zeitraum verwendet wird.

Content-based Filtering Systems.

Bei inhaltsbasierten Systemen enthält ein User Profil die inhaltliche Beschreibung von Gegenständen (zB Web Dokumente mit Hilfe von Begriffen), die den Benutzer interessierten. Dabei wird eine Menge an Eigenschaften von diesem Gegenstand gespeichert, die diesen charakterisieren. Empfehlungen werden mit einer Übereinstimmung der Daten im Profil mit jenen Gegenständen, die der Benutzer noch nicht betrachtet hat, gemacht. Das Benutzerprofil beinhaltet gewichtete Begriffsvektoren (zB basierend auf dem TF.IDF termweighting Model [29]).

Collaborative Filtering Systems.

Kollaborative Filterungssyteme versuchen die Nachteile der beiden anderen Ansätze auszumerzen. Der Fokus liegt hierbei nicht an den Inhalten im Web, sondern vielmehr auf das Finden von Ähnlichkeiten zu anderen Benutzern. Somit wird herausgefunden, wer noch an den selben Dingen interessiert ist, wie der Benutzer. Dadurch liegt der Fokus nicht an dem *WAS* sondern vielmehr an dem *WER*. Hierfür ist man auf die Anzahl der anderen Benutzern und deren bekannten Eigenschaften angewiesen.

2.2.4 Benutzerprofile

Um einen Adaption in einem Softwaresystem zu ermöglichen, müssen spezielle Daten vom Benutzer gesammelt und in einem Benutzermodell (*User Model*) gespeichert bzw. verwaltet werden [21]. Allgemein gesprochen, ist ein Benutzerprofil eine Repräsentation von Informationen über einen individuellen Benutzer [9]. Das Wissen über den Benutzer dient dem System, die angebotenen Informationen an dessen Bedürfnisse anzupassen.

2.2.5 User Profiling

Nach [17] beinhaltet der User Profiling Prozess drei Hauptphasen, welche in Abbildung 2.2 zu sehen sind. Die erste Phase konzentriert sich auf die Sammlung der Informationen über den User. In der zweiten Phase liegt der Fokus auf den Aufbau des Profils. Die finale Phase präsentiert die personalisierten Informationen in einer Anwendung. Im folgenden soll ein Überblick über diese Phasen gegeben werden.

2.2.6 Datengewinnung

Um ein Benutzerprofil aufzubauen, müssen Informationen über den Benutzer gesammelt werden. Dies kann entweder *explizit* durch direkte Benutzerein-

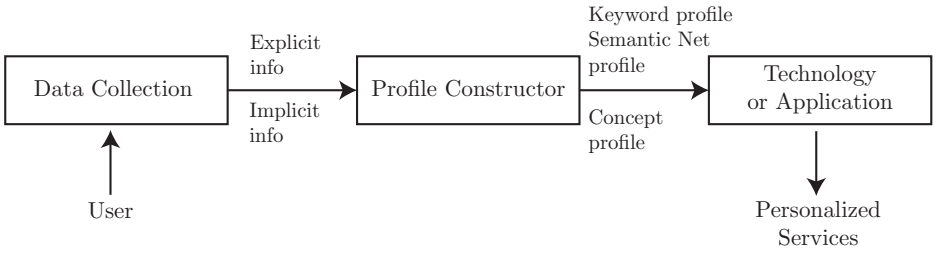

Abbildung 2.2: User Profiling Prozess nach [17].

gabe oder *implizit* mit Hilfe von automatisierten Prozessen oder Agenten, die den Benutzer beobachten, geschehen. Implizit aufgebaute Profile werden auch als *dynamisch* bezeichnet. Dabei wird Personalisierung bei jeder Interaktion mit der Seite durchgeführt und das Benutzerprofil erweitert. Ganz im Gegensatz zu *statischen* Profilen, wo die Informationen pro Benutzer Session gleich bleiben. Wenn bei dynamischen Profilen zusätzlich der Zeitfaktor betrachtet wird, unterscheidet man auch noch zwischen *long-term* - langfristige Interessen, die sich eher selten ändern - und *short-term* - aktuell kurzfristige Interessen.

Methoden zur Datenermittlung

Zu Beginn ist es wichtig, dass der Benutzer vom System wiedererkannt werden kann. Dafür muss dieser eindeutig identifizierbar sein. Hierfür gibt es unterschiedliche Arten, wie das System aufgebaut sein kann. Zum einen kann die gesamte Personalisierung mittels einer Desktop-Applikation durchgeführt werden, worin die Identifikation des Benutzers sichergestellt ist. Weiters kann das System einen Anmelde-Mechanismus verwenden, um den Benutzer zu bestimmen. Alternativ fragt das System fragt die IP Adresse oder Cookie/-Session ID ab [27].

Für die Speicherung der IP Adresse, Cookies und Sessions ID ist vom Benutzers keine Aktionen erforderlich. Diese Information wird vom System selbst beschafft. Oftmals wird zusätzlich eine Anmeldung mit Benutzername und Passwort integriert, da dadurch auch ein Benutzer mit dem eigenen Profil an mehreren Computern die Personalisierung anwenden kann, und auch über mehrere Sessions hinweg, ohne Daten zu verlieren. Für die anderen Varianten (Desktop-Applikation, Proxy Server) muss vom Benutzer eine Aktion getätigt werden, indem entweder eine Applikation oder ein spezieller Server installiert bzw. konfiguriert werden muss. Sie bieten allerdings die sicherste Kontrolle für die Erkennung des Benutzers [17].

Wie bereits angesprochen, gibt es implizite und explizite Methoden, wie Informationen gesammelt werden können. Diese Methoden werden auch in Kombination angewendet.

Explizite Methoden: bauen auf aktiven Interaktionen mit dem Benutzer, in der Regel über Formulare, auf. Dabei werden meistens demographische Daten, Interessen oder Meinungen abgefragt. Insbesondere Fakten wie Beruf, Geburtstdatum etc. können nur auf explizite Weise gewonnen werden.

Problematisch bei diesem Ansatz ist der zusätzliche Zeitaufwand für den Benutzer, welcher die Wartung der Daten manuell durchführen muss. Außerdem ist man angewiesen darauf, dass die Daten korrekt und vollständig angegeben werden. Solche Profile bleiben meist unverändert und decken sich oftmals nicht mit den aktuellen Interessen, da es für den Benutzer zu viel Zeit in Anspruch nehmen würde, die Daten regelmäßig zu warten [17].

Oftmals wird diese Methode verwendet, um Inhalte zu personalisieren und interessante Informationen vorzuschlagen. Benutzer bewerten dabei Informationen (wie zum Beispiel Filme) auf einer linearen Skala. Neue Objekte werden entweder über das kollaborative oder inhaltsbasierte Filtern vorgeschlagen. Ein Vorteil dieser Methode ist, dass für manche Benutzer das Mitteilen und das Veröffentlichen von eigenen Meinungen/Bewertungen genießen [17].

Implizite Methoden: werden für den Benutzer unsichtbar im Hintergrund durchgeführt, während der Benutzer das System verwendet. Der Vorteil bei dieser Methode ist, dass man auf keine aktive Interaktion des Benutzers angewiesen ist. Hierfür gibt es unterschiedliche Technologien, die verwendet werden können, um automatisch die Informationen und Interessen des Benutzers zu erhalten: Desktop Agenten, Proxy Server, Browser Cache, Browser Agenten oder Suchmaschinen-Logs oder Web-Log-Mining [17].

Wenn Daten mit dem Browser-Cache überwacht werden, können sämtliche Aktivitäten des Benutzers innerhalb des Browsers ermittelt werden. Dabei werden einerseits die Daten des Benutzers aus der Browsing History gesammelt, welche Webseiten (URL) besucht wurden, andererseits kann auch auf die Verweildauer, oder die Klicks auf Hyperlinks kann zugegriffen werden.

Ein Nachteil ist, dass die Daten in regelmäßigen Abständen übermittelt werden müssen, ansonsten weicht das Wissen über den Benutzer von den tatsächlichen Interessen ab.

Entweder muss der Benutzer dem System die Daten in regelmäßigen Abständen übermitteln, damit das Profil aufgebaut bzw. erweitert und aktuell gehalten wird, oder der Benutzer installiert einen Proxy Server, der die Aktivitäten automatisch verfolgt.

Vorteile von dieser Technologie ist, dass der Benutzer keine zusätzliche Software installieren muss. Außerdem ist eine umfangreiche Datenbasis vorhanden. Wenn zusätzlichen ein Anmelde-Mechanismum verwendet wird, kann der Benutzer auch das System auf mehreren Computern anwenden [17].

Eine weitere Variante stellen so genannte Software-Agenten dar, welche

entweder in Form eines Desktop-Agenten oder Browser Agenten realisiert werden können. Entweder können Daten in Echtzeit innerhalb von Desktop-Anwendungen personalisiert werden, oder Daten in einem Browser zum Beispiel für die Personalisierung von Webseiten. Desktop-Anwendungen haben den Vorteil, dass auch auf lokale Daten des Benutzers zugegriffen werden kann. Bei Browser-Agenten kann auf alle Aktivitäten des Benutzers zugegriffen werden, während dieser sich im Web bewegt. Zusätzlich zu der URL, der Verweildauer und den Klicks auf Hyperlinks können Browser-Agenten auch auf weitere Aktionen, wie zB Bookmarking oder heruntergeladene Links/-Dateien zugreifen.

Im Gegensatz zur Browsing History, wo bereits eine Menge an Daten hinterlegt sind, benötigt dieser Ansatz Zeit, um genügend Daten vom Benutzer gesammelt zu haben, um eine adäquate Personalisierung durchführen zu können. Außerdem muss eine zusätzliche Software installiert und auch vom Benutzer verwendet werden [17].

Diese Problematik mit der zusätzlichen Software stellt sich allerdings nicht bei reinen serverseitigen Ansätzen, wie Web-Log-Mining oder Suchmaschinen-Logs. Die Suchverläufe dienen als Quelle für die Daten des Benutzerprofils und werden vom Server verarbeitet und in das Benutzerprofil aufgenommen. Oftmals wird dies in Kombination mit einer personalisierten Suche verwendet. Die Menge an Daten ist allerdings eingeschränkt, da man nur Zugriff auf die Suchanfrage hat und nicht auf einen gesamten Text einer Webseite, wie bei den meisten anderen Ansätzen [17].

2.2.7 Aufbau von Benutzerprofilen

Benutzerprofile können entweder mit Hilfe von Schlagwörtern (*Keyword Profiles*), als semantisches Netzwerk (*Semantic Network*) oder als Begriffshierarchie (*Concept Profile*) aufgebaut werden. Wie der Aufbau solcher Profile genau umgesetzt werden kann, wird im Kapitel 3 veranschaulicht. In den nachfolgenden Abschnitten wird ein Überblick über die Eigenschaften dieser Profilarten gegeben.

Keyword Profiles

Die am weitesten verbreitete Variante für die Repräsentation von Benutzerprofilen ist ein Keyword Profile, dass aus einer Menge von gewichteten Schlagwörtern besteht. Dabei werden diese Schlagwörter aus (Web) Dokumente mit Text-Mining-Techniken extrahiert, die vom Benutzer betrachtet wurden. Für die Gewichtung wird oft das Tf-idf-Maß aus dem Information Retrieval verwendet, welche die Vorkommenshäufigkeit von Termen in Dokumenten beschreibt. Dieser Vorgang kann entweder implizit mit Hilfe von Text-Mining-Techniken automatisch von Webseite-URLs extrahiert oder explizit über den Benutzer übermittelt werden. Ein Keyword repräsentiert ein

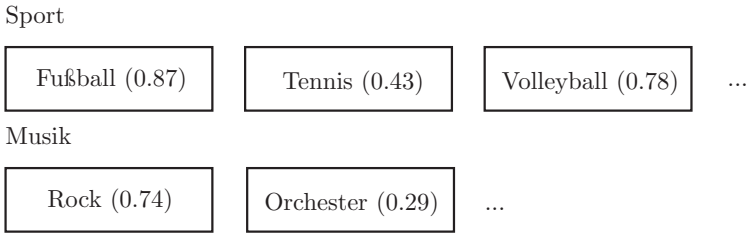

Abbildung 2.3: Keyword Profile in Anlehnung an Gauch et al. [17].

Thema, worüber die Webseite handelt. Die einzelnen Keywords werden mit einer Gewichtung versehen, welche den Interessen- oder Kenntnisgrad als numerischen Wert widerspiegelt. Außerdem können sie auch einer Kategorie zugewiesen werden, um eine Darstellung von Interessenschwerpunkte abzubilden.

Das Hauptproblem bei dieser Variante ist, dass zwischen den unterschiedlichen Dokumenten die extrahierten Wörter übereinstimmen müssen. Dafür benötigt man von Seiten des Benutzers Rückmeldung, welchen Themen in einem Dokument diskutiert werden, da ansonsten Mehrdeutigkeiten oder Synonyme schwer zu unterscheiden sind [17]. In Abbildung 2.3 ist exemplarisch abgebildet, wie ein Keyword Profile aussehen kann.

Semantic Network Profiles

Eine weitere Form der Schlagwort-Profile stellen Profile basierend auf semantischen Netzen dar. Bei dieser Art von Profilen werden die Schlagwörter in einzelnen Knotenkonzepten abgebildet und mit anderen in eine Verbindung gesetzt. Sowohl die Schlagwörter als auch die Verbindung erhält eine Gewichtung. Die Knoten können Überbegriffe oder Klassen darstellen, die mit assoziierte Begriffe oder Unterbegriffe mittels Kanten verbunden sind. Der Vorteil von diesem Ansatz ist, dass besser mit Mehrdeutigkeit und Synonymen der natürlichen Sprache umgegangen werden kann und es einfacher ist, neue Begriffe in das Netz mit einem Bezug auf assoziierte Begriffe zu integriert. Dafür wird oft auf vorgefertigten Worthierarchie wie zB WordNet zurückgegriffen. Im einfachsten Fall werden Begriffe, die gleichzeitig in einem Dokument vorkommen, miteinander verbunden [17]. Abbildung 2.4 stellt exemplarisch dar, wie ein Semantic Network Profile aussehen kann.

Concept Profiles

Ähnlich zu den Semantic Network Profiles beinhalten Concept Profile ebenfalls gewichtete Begriffe als Knoten und Verbindungen zu anderen Knoten. Der Unterschied liegt darin, dass die Knoten mit abstrakten Themen oder Konzepten abgebildet werden. Hierbei wird auf bestehende Begriffshierarchi-

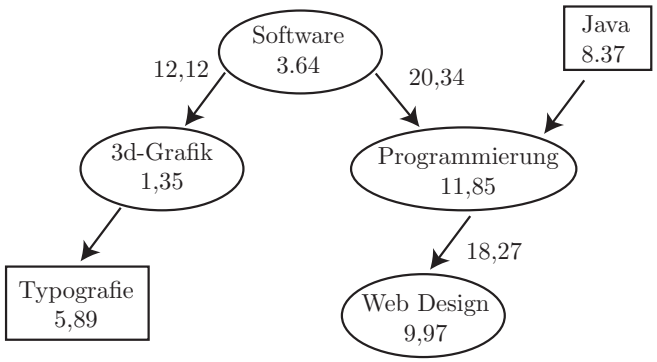

Abbildung 2.4: Semantic Netzwork Profile in Anlehnung an Gauch et al. [17].

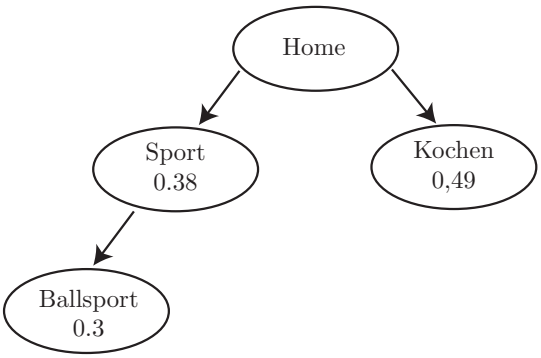

Abbildung 2.5: Concept Profile in Anlehnung an Gauch et al. [17].

en aufgebaut, wie zB ODP oder DBpedia, und kein individuelles Begriffsnetzwerk erstellt. Webseiten werden also analysiert und in eine Hierarchieebene eingegliedert. Dadurch kann die Übereinstimmung zu einem definierten Vokabular geschaffen werden, wodurch eine Generalisierung durchgeführt und eine Erweiterung der Interessengebiete mit Hilfe von Text Klassifikation und der Integration von verwandten Bereichen ermöglich wird. In der einfachsten Form besitzen Begriffe in einem Concept-Profile eine *ist-ein* oder *hat-ein* Beziehung. Dadurch wird erreicht, dass weitere Interessengebiete und verwandte Bereiche in das Profil einbezogen werden. In Abbildung 2.5 wird ein Auszug eines Modelles dargestellt [17].

Kapitel 3

Related Work

Im folgenden Kapitel werden verwandte Arbeiten zum Thema Benutzerprofile und Semantic Web erläutert.

3.1 Extraktion der Benutzerinformation

In der Literatur werden unterschiedliche Datenquellen für Benutzerprofile verwendet, wie zB Suchanfragen [16, 23], Chronik des Browsers mit den besuchten Webseiten [30], Bookmarks [14], Emails und lokal gespeicherte Dokumente[13] oder Daten von Web Communities für kollaborative Informationsfilterung.

3.1.1 Methoden zur Datenermittlung

Als Methode für die Datenermittlung gibt es wie im Abschnitt 2.2.6 erläutert, zwei Ansätze: explizit oder implizit.

Viele Ansätze führen eine explizite Datenermittlung durch, um das Interface anzupassen und somit Informationen leichter für den Benutzer aufzubereiten. MyYahoo[1] erfragt vom Benutzer explizit dessen Präferenzen. Diese werden in einem Profil gespeichert und dafür verwendet, um den Inhalt der Webseite anschließend anzupassen. Google News[2] lässt den Benutzer ebenfalls mit Hilfe einer Skala bewerten, welche Themengebiete gefiltert werden sollen, indem die unterschiedlichen News-Kategorien mit *Selten*, *Gelegentlich*, *Manchmal*, *Häufig*, *Immer* klassifiziert werden.

Auch NetFlix[3] oder IMDb[4] basieren auf expliziten Benutzerbewertungen von Filmen.

Ein bereits altes Projekt von Syskill & Webert [25] empfehlen Informationen basierend auf einer expliziten Benutzerbefragung. Der Benutzer bewertet

[1] http://my.yahoo.com/
[2] http://news.google.at/
[3] http://www.netflix.com/
[4] http://www.imdb.com/

Links auf einer Webseite, woraufhin weitere Links empfohlen werden.

Chirita et al. [11] lässt Benutzer aus der DMOZ Ontologie Knoten auswählen, welche für die Personalisierung berücksichtigt werden sollen.

Explizite Ansätze sind allerdings auf längere Sicht nicht zielführend, da Benutzer nicht gewillt sind, diesen Zusatzaufwand immer wieder zu betreiben, auf Grund von der Bewahrung der Privatsphäre [17].

Matthijs & Radlinski [23] haben ein Firefox Addon namens *AlterEgo* entwickelt, um Zugriff auf die Chronik im Browser zu bekommen. Sobald eine Webseite angesteuert wurde, wird dessen URL, Verweildauer und eine ID des Benutzers übermittelt. Die wichtigsten Begriffe werden extrahiert und mit Hilfe von TF-IDF gewichtet und ins Profil gespeichert. Solch implizite Verfahren werden sehr oft angewendet [10, 11, 15, 18, 20, 22, 32]

3.2 Aufbau von Benutzerprofilen

Ältere Ansätze erstellen Benutzerprofile aus Dokumenten mit den am häufigsten vorkommenden Wörtern und stellen die Interessen im Profil als reine Schlüsselwörter-Modelle mit einer Gewichtung dar. Allerdings geht bei dabei die semantische Bedeutung zwischen Begriffen verloren und folgende Probleme entstehen dadurch [28, 31]:

- Irrelevante Wörter: Innerhalb einer Webseite befinden sich oftmals Wörter, die semantisch nicht relevant sind.
- Mehrdeutigkeit und Synonyme: Wörter können die selbe Bedeutung, oder mehrere Bedeutungen haben. Im Profil muss Klarheit herrschen, was die Begriffe tatsächlich im Text ausgesagt haben.
- Viele unterschiedliche Begriffe befinden sich im Profil nach längerer Anwendung. Nur die aktuell relevanten Interessen sollen allerdings für die Personalisierung von Informationen herangezogen werden.

Um diese Probleme zu beseitigen, werden existierende Worthierarchien oder Wissensdatenbanken für die Ermittlung der Interessen inkludiert. Einige Ansätze verwenden vordefinierte Ontologien für den Aufbau von Benutzerprofilen, wie zum Beispiel DMOZ oder Wikipedia [11].

Semantic Web Technologien können ebenfalls Unterstützung bieten. Es gibt Ansätze, welche die FOAF-Ontologie[8] verwenden, um Benutzerinformationen in einem für Maschinen lesbarem Format zu speichern [8]. Dies bringt den Vorteil, dass die Daten wiederverwendbar und in einer standardisiert Form im Web verfügbar sind [7].

Chirita et al. [11] verwendet Knoten von DMOZ für den Aufbau des Benutzerprofils, um als Basis für eine personalisierte Suche zu dienen. Der Benutzer gibt seine eigenen Interessen explizit mit Hilfe der Auflistung aus der DMOZ Datenbank an. Dadurch werden die von DMOZ inkludierten Daten eingegrenzt. Die ermittelten Interessen werden in den jeweiligen Algorithmen von Suchmaschinen berücksichtigt. Als Basis für die Berechnung von

Ähnlichkeiten zwischen Konzepten werden verschiedene Algorithmen angewandt, die zum Beispiel auf die Messung der Distanz zwischen zwei Knoten beruht. Dieser Ansatz besagt, je weiter auseinander zwei Knoten liegen, desto weniger haben sie miteinander zu tun, da sie wenige Konzepte gemein haben.

Der Baum von DMOZ ist sehr groß und in den meisten Fällen decken die Interessen eines Benutzers nur einen kleinen Teil davon ab. Außerdem fehlen DMOZ oftmals spezifische Interessen. Ein weiteres Problem ist das Finden einer Übereinstimmung. Dafür müssen Klassifikatoren für jeden einzelnen Knoten gebildet werden.

Andere Ansätze verwenden Wikipedia für die Textkategorisierung. Gabrilovich et al. [15] bildet die besuchten Webseiten mit Hilfe von Wikipedia Konzepten ab. Die Ähnlichkeit zwischen einem extrahierten Begriff und einem passenden Wikipedia Konzepte wird gesucht und als Gewichtung verwendet. Als Basis für die Bestimmung der passenden Übereinstimmung werden die Verlinkungen innerhalb eines Artikels verwendet.

Wikipedia eignet sich gut für die Kategorisierung von Ressourcen, da es eine große Palette an Interessengebiete abdeckt. Außerdem befindet sich hinter einem Wikipedia Konzept weitere Informationen, welche über eindeutige URIs miteinander verknüpft sind.

Bei Ramanathan und Kapoor [28] wird ein hierarchisches Benutzerprofil ebenfalls mit den Konzepten aus Wikipedia aufgebaut. Zuerst werden Webseiten und Dokumente dem Wikipedia Konzept zugeordnet und anschließend anderen Konzepten im Profil zugeordnet, entweder als Kind- oder Elternelement.

Ein hierarchisches Benutzerprofil als Baumstruktur wird von Kim et al. [20] verwendet, um die Interessen abzubilden. Dabei wird zwischen allgemeine Interessen, die an der Wurzel angesiedelt sind (langfristige Interessen) und jene Themen an den Knoten, welche den Benutzer generell einmal interessiert haben (kurzfristige Interesse) unterschieden. Dieser Ansatz wird *User Interest Hierarchy* (UIH) genannt. Als Daten werden dabei besuchte Webseiten verwendet. Die am häufigst vorkommenden Wörter pro Dokument werden extrahiert und anschließend mit Ähnlichkeitsalgorithmen in der Hierarchie zu einem Cluster zusammengeführt. Dadurch wird die Zugehörigkeit zwischen Begriffen zu weiteren Themengebieten ermöglicht.

Xu et al. [32] wenden einen ähnlichen Ansatz an, indem zuerst die oft vorkommenden Begriffe aus einem Dokument extrahiert und anschließend als Knoten dargestellt werden. Die Dokumente wiederum werden als Mengenpaare diesen Knoten hinzugefügt. Anschließend werden ähnliche Begriffe analysiert.

Nach Chan [10] wurde ein adaptiver personalisierter Web Browser entwickelt, der rein auf implizite Weise die Informationen des Benutzers sammelt. Es wurde ein Messwert entwickelt, der einer besuchten Webseite einen Wert zuweist, der aussagt, in wie weit der Benutzer daran interessiert ist. Bei die-

sem Ansatz wird ein Benutzerprofil mit zwei Komponenten definiert: *Page Interest Estimator* (PIE) und einem *Web Access Graph* (WAG). Basierend auf den *PIE*, der aus vom Verhalten des Benutzers erlernt wird, werden die Interessen berechnet. Der *WAG* beinhaltet die besuchten Webseiten mit den gesammelten Mustern und mit Hilfe des *PIE* wird deren Relevanz errechnet. Um die Interessen zu sammeln, wurden folgende Quellen berücksichtigt: Browsing History, als Lesezeichen abgespeicherte Webseiten, besuchte verlinkte Seiten auf einer Webseite und Server Log Dateien (Größe der Seite, Zeitpunkt für die Abfrage der Aktualität, IP Adresse, URL).

Grcar et al. [18] entwickelten ein System, welches im Internet Explorer integriert wurde. Darin wird ein Benutzerprofil verwaltet, indem automatisch Themen, die den Benutzer interessieren, in Form einer Ontologie aufgebaut werden. Ähnlich wie bei [20] wird ebenfalls zwischen langzeitigen und kurzzeitigen Interessen unterschieden.

Lu et al. [22] entwickelten ein Empfehlungssystem für Tweets auf Twitter. Von den Tweets werden Wikipedia Konzepte entnommen und zu einem gewichteten Vektor zusammengebaut. Die Gewichtung basiert auf der Ähnlichkeit zwischen Konzept und Text. Der Wikipedia Graph wird zufällig durchlaufen und mit Hilfe eines Ähnlichkeitsalgorithmus passende Konzepte ausgewählt und hinzugefügt, um die Interessen des Benutzers zu erweitern und die Empfehlung zu verbessern. In diesem Ansatz liegt der Fokus auf die Verlinkungen innerhalb des Wikipedia-Artikels.

Kapitel 4

Entwurf

Das folgende Kapitel gibt einen Überblick über die Konzeption des entwickelten Ansatzes und beinhaltet weiteres Vorwissen für das darauffolgende Kapitel der Implementierung einer Browser-Erweiterung zu dem Aufbau eines Benutzerprofils.

4.1 Ausgangssituation

Auf Grund von den immer größer werdenden Mengen an Informationen ist es für Benutzer schwer, die relevanten von den nicht relevanten Informationen zu unterscheiden. Für einen Benutzer interessante Informationen gehen oft verloren oder werden übersehen. Um hierfür eine Unterstützung zu bieten, benötigen Informationssysteme Wissen über den Benutzer in strukturierter Form, sodass diese in den Personalisierungsprozess integriert werden können. Da dies ein Grundstein für die Personalisierung darstellt, wird in den folgenden Abschnitten ein Konzept erläutert, wie Benutzerprofile mit Semantic Web Technologien aufgebaut werden können.

Folgende Anforderungen sind auf Grund von der Problemstellung definiert worden:

- Implizite automatisierte Ermittlung der Benutzerinformationen,
- Keine Unterbrechungen/Störungen für den Benutzer,
- Aufbau einer breiten Wissensbasis über den Benutzer,
- Entwicklung einer Grundlage in Form von Interessen des Benutzers, welche in Filterungs- und Personalisierungssysteme integriert werden kann,
- Bewahrung der Privatsphäre – Möglichkeit zur Deaktivierung.

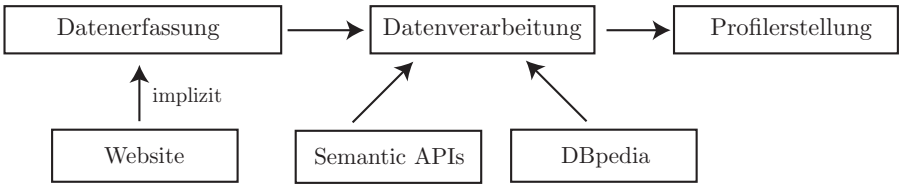

Abbildung 4.1: Schritte des Personalisierungsprozesses in Anlehnung an Gauch et al. [17].

4.2 Spezifikation

Anhand der Anforderungen sind verschiedene Bereiche ausgearbeitet worden, um diese zu erfüllen. Abbildung 4.1 stellt den Personalisierungsprozess grafisch dar, wie Wissen über den Benutzer generiert und in Form von Interessen in einem Profil gespeichert werden.

Der in Abbildung 4.1 graphisch beschriebene Personalisierungsprozess wird in folgender Liste nochmals angeführt, welche die weitere Struktur der Arbeit beschreibt:

1. **Datenerfassung** auf Grund vom Benutzer besuchter Webseiten,
2. **Datenverarbeitung** in Form von der Extraktion von Schlüsselwörter und weiterer Anreicherung dieser Daten,
3. **Datenspeicherung** in hierarchischer Form innerhalb eines Benutzerprofils.

4.3 Datenerfassung

Wie bereits in Abschnitt 2.2.6 erläutert, ist die Sammlung von Wissen über den Benutzer die Grundlage für den Personalisierungsprozess. Als Basis für diesen Schritt werden die im Web betrachteten Webseiten herangezogen. Es wird davon ausgegangen, dass die Seiten, die der Benutzer im Web betrachtet, dessen Interessen widerspiegeln.

4.3.1 Gewählte Methode zur Datenerfassung

Für die Datenerfassung wird eine implizite Methode gewählt, da bereits unterschiedliche Studien gezeigt haben, dass Benutzer unwillig sind, persönliche Daten selbst in regelmäßigen Abständen einzugeben (siehe Kapitel 3).

Für die Ermittlung der besuchten Webseiten wird eine Google Chrome Browser Erweiterung implementiert, welche innerhalb des Google Chrome Browsers läuft. Die Variante der Browser Erweiterung wird gewählt, da somit Zugriff auf die benötigte Benutzerinformation - besuchte Webseite - ermöglicht und eine einfache Nutzung unterstützt wird. Um die Privatsphäre

zu bewahren, muss der Benutzer schnell die Möglichkeit haben, die Funktionalität zu deaktivieren und immer einen Überblick über den aktuellen Status besitzen. Da eine Erweiterung innerhalb des Browsers in der Taskleiste angesiedelt ist, werden diese Anforderungen erfüllt. Der Browser ist eine Einschränkung für die Nutzung es Systems, allerdings muss keine eigenständige Software für das Surfen im Internet verwendet werden. Der Browser wird auf Grund von technische Voraussetzungen gewählt, die sowohl für eine leichte Installation, aber auch für den schnellen und einfachen Zugriff auf die benötigten Informationen ermöglichen wie zB *Content Scripts*. Mehr dazu im Abschnitt 5.3.1. Google Chrome ist weltweit der führende Browser. Auch in Österreich hat er bereits Platz 2 erreicht und ist stetig am Steigen[1].

Die Google Chrome Erweiterung läuft innerhalb des Browser und übermittelt laufend im Hintergrund die besuchten Webseiten, ohne des Benutzers angewiesen zu sein oder ihn zu unterbrechen. Um dem Benutzer auch die Möglichkeit zu geben, mitzubestimmen, kann dieser auch selbst Interessen angeben und löschen.

Ein Login in Kombination mit der Information aus der Browser-Session ermöglicht die Identifikation des Benutzer. Dadurch können mehrere Benutzer die Anwendung an einem Computer verwendet und die Generierung der Interessen wird nicht an einen Computer gebunden.

Die Verwendung der Erweiterung wird mit einem Klick in der Taskleiste innerhalb des Browsers aktiviert bzw. deaktiviert. Die Benutzer entscheiden somit selbst, ob die Interessen analysiert und gespeichert werden sollen oder nicht.

4.3.2 Herangehensweise

Bei jedem Aufruf einer Webseite wird die URL analysiert. Für die Textextraktion ist eine existierende Semantic API eingesetzt worden namens Zemanta[2], welche auf die Extraktion von Entitäten und Schlüsselwörtern spezialisiert ist. Diese API liefert nicht nur die am häufigsten vorkommenden Schlüsselwörter, sondern führt eine semantische Analyse des Textes durchführt, womit auch semantisch ähnliche Begriffe ermittelt werden. In Abschnitt 4.6 werden die verwendeten APIs kurz erläutert.

Nach einer Studie von Bauer [2] ist Zemanta sehr gut geeignet für die Extraktion von Schlüsselwörtern[3]. Deswegen wird diese auch hier für die Extraktion verwendet. Der einzige Nachteil dieser API ist, dass nur die englische Sprache unterstützt wird. Deswegen weren nur englische Begriffe bei der Implementierung berücksichtigt.

[1] http://gizmo247.com/browserstatistik-google-chrome-zieht-davon/, zuletzt aufgerufen am 29.09.2013

[2] zemanta.com

[3] Eine Demonstration der Funktionalität befindet sich unter http://www.zemanta.com/demo/.

Zemanta ist über ein Webservice über einen HTTP-Request erreichbar und nimmt als Parameter einen Klartext entgegen, um brauchbare Entitäten zu analysieren. Um den Klartext von einer Webseite zu erhalten, wird die AlchemyAPI verwendet, welcher den Text vor der Analyse optimiert. Wenn mittels HTTP-Request der bereinigte Text Zemanta übergeben wird, wird eine Antwort in .json-Format in folgendem Format zurückgeliefert.

```
...
{
  "keywords": {
    "keyword": [
      {
        "confidence": 0.506297,
        "name": "Mars",
      },
      {
        "confidence": 0.296248,
        "name": "Phoenix",
      },
      ...,
    ]
  }
} ...
```

Um aus diesen einzelnen Begriffen mehr Informationen zu generieren und ein hierarchisches Benutzerprofil mit sowohl spezifischen, als auch allgemeinen Themengebieten aufbauen zu können, werden diese extrahierten Begriffe noch einer weiteren Analyse und Anreicherung unterzogen.

4.4 Datenverarbeitung

Die Anreicherung der extrahierten Schlüsselwörter wird mit dem Wissen und der Struktur von Wikipedia ermittelt. Wikipedia wird gewählt, da es die größte Wissensdatenbank besitzt und die Daten in strukturierter Form beinhaltet. Außerdem verfügt es über eine hierarchische Struktur, worin die einzelnen Bereiche in gleichem Ausmaß ausgeprägt sind und nicht bei einigen Bereichen kaum Informationen vorhanden sind [15].

Die Struktur und das Wissen von Wikipedia wird in diesem Ansatz verwendet, um das Benutzerprofil aufzubauen. DBpedia stellt die Informationen in RDF-Form öffentlich zur Verfügung. Es beinhaltet eine große Menge an unterschiedlichem Wissen in Form von Artikeln und Verknüpfungen zu anderen Informationen, auf die mit eindeutigen URIs zugegriffen werden kann. Für den Aufbau des Profils werden die Wikipedia-Artikel als Entitäten und zusätzlich die Kategorie-Struktur verwendet, um ein hierarchisches

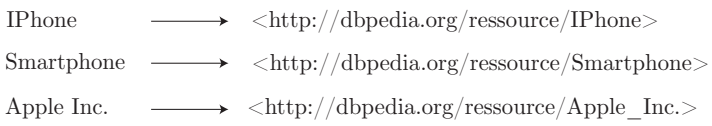

Abbildung 4.2: Zuordnung von Entitäten mit DBpedia-URI.

Abbildung 4.3: *dc:subject* zeigt auf direkte Kategorien einer Entität.

Konstrukt an Begriffen zu schaffen. Die von der API extrahierten Schlüsselwörter dienen für die Repräsentation der spezifischen Interessen und als Basisdaten für die weitere Verarbeitung. Die zusätzlich analysierten Kategorien reichern automatisch diese Schlüsselwörter an, sodass dieser genauer beschrieben und das Benutzerprofil erweitert wird. Probleme, wie zB die Mehrdeutigkeit oder verwandte Begriffe oder Themen werden somit berücksichtigt und unterstützt.

4.4.1 Herangehensweise

Da eine externe Wissensdatenbank verwendet wird, muss zunächst eine Übereinstimmung zwischen dem extrahierten Schlüsselwort und einer passenden Entität (Artikel) von DBpedia gefunden werden, um von den weiteren Informationen im DBpedia Graphen zu profitieren. Hierfür unterstützt die DBpedia Spotlight API, die aus einem Text bzw. Begriff eine passende DBpedia-Ressource annotiert und als Antwort die eindeutige URI der DBpedia Entität zurückliefert. Abbildung 4.2 zeigt grafisch, wie sich die Zuweisung eines Begriffs zu einer DBpedia Entität vorzustellen ist. Diese eindeutige URI ermöglicht, dass eine Brücke von der DBpedia Entität zu weiteren Informationen aufgebaut werden kann, wie in Abbildung 4.3 dargestellt. Das Attribut *dc:subject* ist die Verbindung von einem Artikel zu den zugewiesenen Kategorien, die wiederum mit einer eindeutigen URI ausgezeichnet und mit weiteren Informationen verknüpft ist. Die genaue Beschreibung von *dc:subject* folgt in diesem Kapitel im Abschnitt 4.5.1. Diese Information in Form von Kategorien wird für die Anreicherung verwendet, da diese mit dem Ursprungsbericht (der Entität) in DBpedia über *dc:subject* verknüpft sind und somit Beschreibung bzw. neue Konzepte damit in Verbindung gebracht werden können.

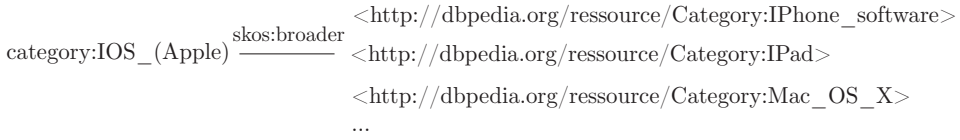

Abbildung 4.4: Das Attribut *skos:broader* zeigt auf übergeordnete Kategorien einer Kategorie.

Tabelle 4.1: *Main topic classifications* aus DBpedia.

Education	Science
Geography	Society
Law	Technology
Belief	Agriculture
Chronology	Arts
Culture	Environment
Health	Life
History	Mathematics
Humanities	Business
Nature	Language
People	Politics

DBpedia zeichnet spezielle Kategorien als Hauptkategorien aus, die mit dem Konzept *Main topic classification*[4] verbunden sind, welche in Tabelle 4.1 gezeigt werden. Da innerhalb des Profils auch allgemeine Interessen dargestellt werden sollen, werden diese Kategorien dafür verwendet und automatisch zu den extrahierten Schlüsselwörtern analysiert. Um eine passende Hauptkategorie zu finden, muss der optimale Weg im DBpedia-Graphen gefunden werden. Somit muss von einer ermittelten Kategorie eines Artikels wiederum dessen verbundenen Überkategorien (siehe Abbildung 4.4) analysiert und verfolgt werden, solange bis eine passende Hauptkategorien gefunden wird. Diese Kategorien werden in weiterer Folge als Zwischenkategorien bezeichnet, welche weitere Informationen zum Ursprungsbegriff anbieten und als mögliche weitere Interessen des Benutzers dienen. Außerdem werden die Verbindungen zueinander innerhalb des hierarchischen Benutzerprofil gespeichert.

Um Zugriff auf die Daten innerhalb von DBpedia zu erhalten, gibt es zwei Möglichkeiten. Mit Hilfe eines Webservices[5] können Entitäten von einer Live-Datenbank über einen Virtuoso SPARQL-Query-Editor mittels SPARQL-Select-Statements abgefragt werden. Dieser Weg wurde zu Beginn der Um-

[4] http://dbpedia.org/page/Category:Main_topic_classifications
[5] http://de.dbpedia.org/sparql

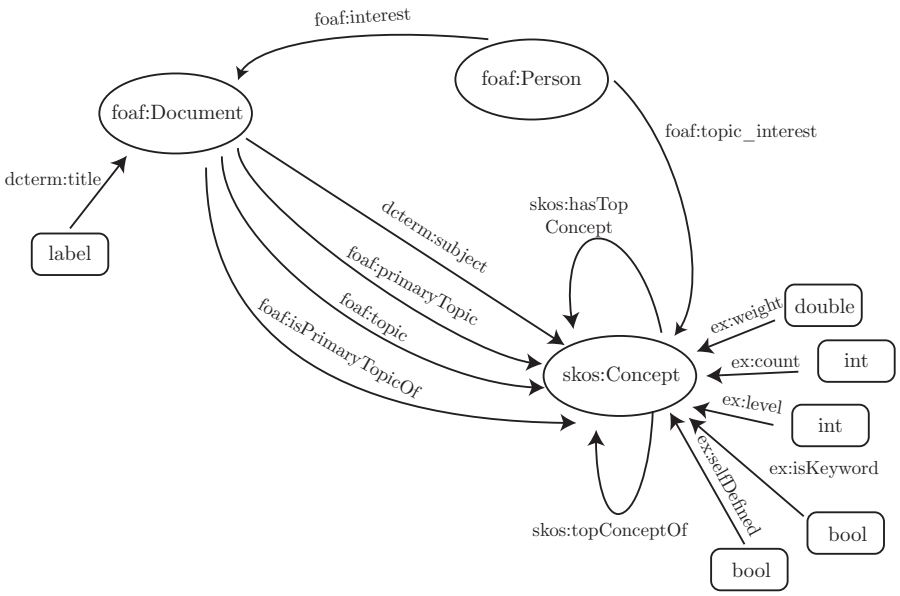

Abbildung 4.5: Konzept des Benutzermodells.

setzung auch verfolgt. Allerdings kam es immer wieder zu Ausfällen der API, wobei die Schnittstelle nicht aufgerufen werden konnte. Außerdem werden viele Requests pro extrahiertes Schlüsselwörter benötigt, bis der Lösungsweg zu einer passenden Hauptkategorie gefunden wird, weshalb es zu langen Wartezeiten geführt hat. Deswegen wurden im Laufe der Entwicklung die DBpedia Daten lokal installiert. Im Anhang A.2 befindet sich eine Installationsbeschreibung dafür.

4.5 Speicherung Benutzerprofil

Im Benutzerprofil werden die beschriebenen Begriffe und Konzepte für einen Benutzer gespeichert. Dieser erhält eine eindeutige URI, wodurch die Konzepte mit der Person verbunden und auch mit anderen Daten im Web in Verbindung gebracht werden können.

Da hauptsächlich Konzepte im Profil gespeichert werden und eine semantische Hierarchie abgebildet wird, wird das Profil in Anlehnung an *Concept Profiles* und *Semantic Network Profiles* entwickelt. Abbildung 4.5 zeigt das entwickelte Benutzermodell, welche als Vorlage für das Benutzerprofil dient.

4.5.1 Bestehende Konzepte

Im Web gibt es bereits einige Strukturen und Modelle, die Informationen rund um eine Person abdecken. In den folgenden Abschnitten wird erklärt, wie bereits existierende Konzepte in das Benutzermodell integriert sind, um die gespeicherten Informationen im Web wiederverwendbar und erweiterbar zur Verfügung zu stellen. DBpedia beinhaltet diese und weitere Konzepte für die Anreicherung und die Verlinkung von Informationen im Web.

FOAF (The Friend of a Friend project)

Die Grundidee des *Friend-of-a-Friend* Projekts[6] ist, dass personenbezogene Informationen im Web mit einer einheitlichen Struktur abgebildet werden. Es baut auf die RDF-Technologie auf und soll Maschinen helfen, Verbindungen in Form einer eindeutig zugeordneten URI zwischen Informationen zu finden. Dies deckt sowohl allgemeine Informationen über Personen, wie Name, Webseite, Alter, uvm. aber auch Interessen und Verbindungen zu anderen Personen ab. Auch Informationen zu Bookmarks, Bewertungen, Publikationen, Mitgliedschaften in Organisation oder berufliche Anstellung können mit dieser Struktur abgebildet werden [8].

Die Struktur von FOAF wird im Benutzerprofil für die Auszeichnung der Personendaten, aber auch für die besuchten Webseiten verwendet. Die verwendeten Attribute sind in der Tabelle 4.2 aufgelistet und in weiterer Form mit dem Präfix `foaf` in Verbindung zu bringen. Die vollständige Auflistung ist im Web einzusehen [47].

Tabelle 4.2: Verwendete Konzepte aus dem FOAF-Vokabular, um die personenbezogenen Informationen und Webseiten zu repräsentieren.

FOAF-Vokabular	Beschreibung
`foaf:name`	Der Name des Benutzers.
`foaf:interest`	Bezeichnung für die besuchten Webseiten.
`foaf:topic`	Die ermittelten Konzepte von einer besuchten Webseite und die Verbindung zwischen besuchter Webseite und Benutzer.
`foaf:topic_interest`	Die ermittelten Konzepte in Verbindung mit dem Benutzer.

[6] http://www.foaf-project.org/

Dublin Core

Dublin Core[7], entwickelt von der *Dublin Core Metadata Initiative*, ist eine Spezifikation für die Beschreibung von Dokumenten, mit dem Ziel, Informationen mit Hilfe von Metadaten einfacher im Web zu finden. Die verwendeten Attribute werden in Tabelle 4.3 erläutert, welche im Laufe dieser Arbeit mit dem Präfix `dcterm` in Verbindung erscheinen [36]:

Tabelle 4.3: Integrierte Konzepte aus dem Dublin Core-Vokabular für die Auszeichnung von allgemeinen Informationen über besuchte Webseiten und Beziehungen zwischen den extrahierten Konzepten und dem Benutzer.

Dublin Core Vokabular	**Beschreibung**
`dcterm:id`	Eine eindeutige Identifizierung für eine Entität (Dokument, Webseite, Person, etc.).
`dcterm:subject`	Eine allgemeine Beschreibung des Inhaltes und wird für die Verbindung zwischen Webseite und Konzept und als Verbindung zwischen Schlüsselwort und ermittelten Konzepten verwendet.
`dcterm:title`	Bezeichnet den Titel der Webseite und den Titel der ermittelten Konzepte.

SKOS (SKOS Simple Knowledge Organization System)

SKOS[8] wird vom W3C für die Organisation von Wissen empfohlen. Es beinhaltet Spezifikationen und Standards, um Wissen in RDF-Form auszuzeichnen, um Informationen einheitlich als Linked Data zur Verfügung zu stellen und austauschbar darzustellen [46]. Die Tabelle 4.4 zeigt die verwendeten SKOS-Elemente und beschreibt deren Verwendung. In weiterer Folge sind diese Eigenschaften mit dem Präfix `skos` verknüpft [44].

4.6 Exkurs: (Semantic) APIs

Im folgenden wird ein grober Überblick über die ausgewählten Semantic APIs gegeben. Eine Semantic API ist eine Schnittstelle, die einen unstrukturierten Text entgegennimmt und diesen semantisch analysiert. Als Antwort, zumeist in XML-, JSON oder RDF-Format, werden unterschiedliche Informationen über den Kontext zurückgegeben, wie zB extrahierte Schlüsselwörter bzw. Entitäten, verwandte Informationen, Kategorien, uvm. [12].

[7] http://dublincore.org/
[8] http://www.w3.org/2004/02/skos/

Tabelle 4.4: Auszug aus dem SKOS-Vokabular für die Darstellung der Interessen und Beziehung zu verwandten Konzepten.

SKOS-Vokabular	Beschreibung
`skos:Concept`	Fundament der SKOS-Spezifikation für die Auszeichnung der ermittelten Konzepte.
`skos:broader`	Eigenschaft für die Darstellung der hierarchischen Beziehung zwischen den extrahierten Entitäten.
`skos:label`	Bezeichner für ein `skos:Concept`
`skos:topConceptOf`	Verbindung zwischen den ermittelten Entitäten zu dem Benutzer
`skos:topConceptOf`	Verbindung zwischen einer Hauptkategorie zu den anderen verbundenen Konzepten basierend auf dem extrahierten Schlüsselwort.
`skos:hasTopConcept`	Verbindung zwischen dem extrahierten Schlüsselwort zu einer ermittelten Hauptkategorie.

Zemanta

Zemanta[9] generiert Metadaten auf Grund eines unstrukturierten Textes. Semantische Analysen extrahieren relevante Schlüsselwörter und Entitäten aus einem Text. Die Informationen sind nicht immer unmittelbar im Text enthalten. Es werden ebenfalls zusätzliche Informationen analysiert, wie zB Bilder oder verwandte Artikel, die im Web über die Linked Data Cloud verfügbar sind. Mit diesen weiteren Informationen können Texte oder Begriffe angereichert werden. Für die Verwendung der API ist eine Registrierung bei Zemanta erforderlich, um einen API-Keys anfordern zu können [50].

Die Analyse der Informationen basiert auf folgenden Mechanismen [51]:

> Basis for Zemanta's suggestions are state of the art algorithms for processing natural language, machine learning, information retrieval and similar.

Dies bedeutet, dass die unstrukturierten Texte mit diversen Datenbanken abgeglichen werden, die für diesen Prozess als Hintergrundwissen fungieren. Mit Hilfe von statistischen Methoden aus dem Bereich Information Retrieval werden verwandte Elemente im Internet gesucht und vorgeschlagen. Informationen können allerdings nur gefunden werden, wenn sich diese in strukturierter Form innerhalb der Linked Data Spezifikation befinden. Folgende Informationen können angefordert werden [51]:

Bilder: Datenbanken wie Wikipedia und Flickr werden auf Übereinstimmung durchsucht, um relevante Bilder zu einem Text zu finden.

[9] http://www.zemanta.com/

Relevante Artikel: Diverse Newsportale und Blogs werden analysiert und als verwandte Artikel zu einem Text angeboten.

Links: Zu einzelnen Entitäten, wie zum Beispiel bei Personen oder Organisationen, werden eindeutige URIs für Entitäten innerhalb von Wissensdatenbanken geliefert, wie zB Wikipedia[10], IMDB[11], MusicBrainz[12])

Keyword Extraktion: Acht relevante, nicht unbedingt unmittelbar explizit im Text vorkommende Wörter oder Phrasen werden zurückgeliefert.

Kategorie: Eine Kategorie, die zum Text semantisch zuordenbar ist, wird analysiert.

AlchemyAPI

Die AlchemyAPI[13] ist wie Zemanta ein Anbieter von semantischen Textanalysesystemen. Es werden unterschiedliche Sprachen unterstützt: Deutsch, Englisch, Französisch, Italienisch, Portugiesisch, Russisch, Schwedisch und Spanisch. Bei der AlchemyAPI kann sowohl ein Text, eine URL oder eine HTML Seite übergeben werden. Die API bereinigt den Inhalt einer Webseite automatisch. Auch für diese API ist eine Registrierung und ein Key erforderlich[14]. Folgende Funktionalität bietet die AlchemyAPI an:

Entity Extraction: Im Text vorkommende Begriffe werden in Entitätstypen eingeordnet wie *Person*, *Company*, *City* und wenn vorhanden mit Objekten aus der Linked Open Data Cloud verknüpft.

Keyword Extraction: Wichtige Wörter und Phrasen werden semantisch analysiert und extrahiert.

Concept Tagging: Hierbei wird das Konzept des übermittelten Inhaltes bestimmt.

Sentimental Analysis: Bei der Stimmungsanalyse wird, sowohl bei dem Text als auch bei den Entitäten und Keywords ein positiver oder negativer Wert nach dessen Stimmung bzw. Meinung zugeteilt.

Document Categorization: Die AlchemyAPI liefert eine von 12 vorhandenen Kategorie zurück.

Language Detection: Eine von 97 verschiedenen Sprachen wird dem Text oder der Seite zugeordnet.

Content Scraping: Bestimmte strukturierte Daten wie zum Beispiel Produkteigenschaften, Beschreibungen oder Kosten werden aus dem Text oder der Webseite extrahiert.

[10] http://www.wikipedia.at
[11] http://www.imdb.com
[12] http://musicbrainz.org
[13] http://www.alchemyapi.com
[14] http://www.alchemyapi.com/api/register.html

DBpedia Spotlight

DBpedia Spotlight[15] ist eine Anwendung, welche automatisch DBpedia Konzepte in einem Text annotiert. Es erkennt den Namen einer Entität und ermittelt für diesen Begriff die eindeutige URI aus DBpedia. Hauptsächlich wird DBpedia Spotlight für Text Annotationen verwendet und kann somit Hintergrundwissen liefern oder Information Retrieval Aufgaben unterstützen. Es gibt unterschiedliche Endpunkte, womit sie als Webservice verwendet werden kann [35]:

Spotting: Erkennt Entitäten/Konzepte von DBpedia aus einem unstrukturierten Text.

Disambiguate: Nimmt bereits annotierte Entitäten und versucht den Begriff mit Hilfe von der DBpedia Datenbank aufzuklären, falls dieser nicht direkt vorhanden ist.

Annotation: Führt beide Aufgaben von *Spotting* und *Disambiguate* gleichzeitig aus.

Candidates: Ähnlich der *Annotation*-Funktionalität, jedoch wird eine gereihte Liste mit allen passenden Entitäten erhalten.

[15] https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki

Kapitel 5

Implementierung

In diesem Kapitel wird basierend auf der konzeptionellen Beschreibung die Implementierung des Benutzerprofils erklärt. Es wird gezeigt, mit welchen technischen Methoden und Überlegungen die Interessen ermittelt und anschließend im Benutzerprofil gespeichert werden.

5.1 Verwendete Technologien

Als Grundtechnologie der Anwendung wird PHP verwendet mit Hilfe einer MySQL-Datenbank, worin die Informationen des Benutzers in RDF-Form gespeichert werden. Eine reine relationelle Datenbank ist nicht zielführend, da zwischen den einzelnen Daten innerhalb der Datenbank eine aussagekräftige Beziehung gespeichert werden muss, um eine semantische Abfrage der Informationen zu ermöglichen. Hierfür dient das Framework *ARC*[1], welches als Unterstützung für den Umgang mit RDF-Daten dient. Dieses Framework ist für Semantic Web Anwendungen in PHP geeignet und ist ein Open-Source Projekt. Es beinhaltet SPARQL-Endpoint-Klassen und einem RDF-Triple-Store innerhalb einer MySQL-Datenbank. Für die lokale Installation der DBpedia Daten wird ein Tomcat-Server mit einem Sesame Triple Store[2] installiert, anstelle der Verwendung des Webservices. Für eine genaue Beschreibung dieser Installation wird auf den Anhang A.2 verwiesen.

Sowohl für die Datenbank des Benutzerprofils, als auch für die DBpedia Datenbank werden SPARQL-Statements formuliert, um die Daten abzufragen bzw. zu speichern. Die verwendeten Präfixe werden zu Beginn vorweggenommen und im Programmcode 5.1 präsentiert. In den folgenden Statements werden diese somit nicht mehr berücksichtigt, da die Präfixe ohnehin immer gleich sind.

[1] https://github.com/semsol/arc2/wiki
[2] http://www.openrdf.org/

Programm 5.1: Präfix-Auszug für alle folgenden SPARQL-Statements.

```
1 @prefix ex: <http://www.example.org#> .
2 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
3 @prefix foaf:  <http://xmlns.com/foaf/0.1/> .
4 @prefix dcterm: <http://purl.org/dc/terms/subject#> .
5 @prefix skos: <http://www.w3.org/2004/02/skos/core#> .
6 @prefix db: <http://dbpedia.org/page/#> .
7
```

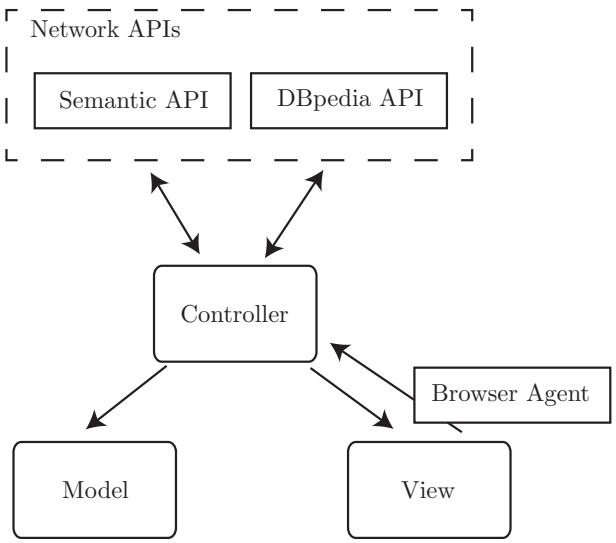

Abbildung 5.1: Architekturmuster der Anwendung in Form von dem Model-View-Controller Prinzip.

5.2 Systemarchitektur

Als Architekturmuster wird das Model-View-Controller Prinzip verwendet, welches in Abbildung 5.1 zu sehen ist.

Ein Controller übernimmt die Logik der Anwendung und verarbeitet Anfragen vom Benutzer innerhalb des Browsers. Genauer gesagt erhält der Controller von der Browser Erweiterung automatisch die extrahierten Benutzerinformationen in Form von der besuchten Webseite. Die Erweiterung dient als Schnittstelle zwischen Controller und dem Benutzer bzw. der View. Die Browser Erweiterung im Browser Google Chrome entnimmt die aktuelle URL der besuchten Webseite von Benutzer und reicht diese über einen Ajax-Befehl zu dem Controller weiter. Dieser analysiert und verarbeitet die relevanten Begriffe basierend auf dem Inhalt der Webseite, welche als Inter-

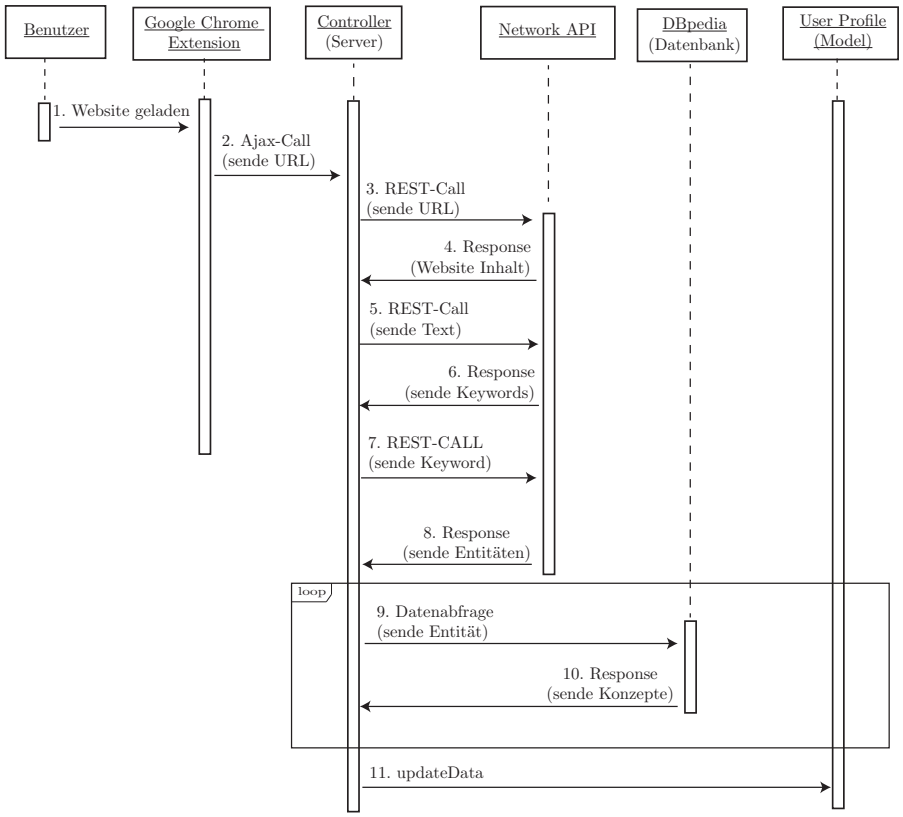

Abbildung 5.2: Der Kommunikationsablauf des gesamten Prozess in Form eines Sequenzdiagramms.

essen in das Benutzerprofil (Model) gespeichert werden. Der beschriebene Ablauf des Prozesses ist in Form eines Sequenzdiagramms in Abbildung 5.2 dargestellt. Die folgende Liste listet die vier Hauptschritte im Prozess erneut auf und zeigt die weiterführende Kapitelstruktur:

1. Extraktion der Benutzerinformation mit einer Google Chrome Erweiterung,
2. Extraktion der Schlüsselwörter mit externen Semantic APIs,
3. Anreicherung der Schlüsselwörter basierend auf DBpedia Kategorien,
4. Speicherung der Daten.

5.3 Extraktion der Interessen

Da die Information vom Benutzer ohne aktiven Einfluss gesammelt werden soll, wird die implizite Variante mit Hilfe eines Browser Agenten imple-

mentiert. In folgenden Abschnitten wird die Umsetzung der Google Chrome Erweiterung beschrieben und anschließend auf die Extraktion der Schlüsselwörter und Anreicherung mit den DBpedia Kategorien eingegangen.

5.3.1 Extraktion der Benutzerinformation

Eine Google Chrome Erweiterung wird verwendet, welche für die Beobachtung des Benutzerverhaltens verantwortlich ist. Mit Erweiterungen innerhalb eines Browsers können dessen Funktionalitäten ergänzt werden, ohne tief in den nativen Source Code des Browser eindringen zu müssen. Für den Browser Google Chrome sind dafür die Technologien HTML, CSS und JavaScript zu verwendet.

Allgemeiner Aufbau

Die folgenden Informationen basieren auf der Developer Dokumentation von Google[3] und zeigen die Implementierungsdetails der Erweiterung. In nachkommender Liste wird die Struktur der Dateien innerhalb der Erweiterung gezeigt:

- *manifest.json* ist die Konfigurationsdatei in .json-Format, welche die Eigenschaften der Erweiterung beinhaltet, wie zB Name der Erweiterung, Beschreibung, Versionsnummer, Permissions, uvm.) (zu sehen im Programm 5.2).
- HTML-Dateien zeigen das User Interface der Extension (zB *popup page* und *option page*).
- JavaScript Dateien decken die Logik ab.

Architektur

Prinzipiell wird unterschieden zwischen *Browser Actions* und *Page Actions*. Letztere sind innerhalb einer bestimmten Seite über die Adressleiste sichtbar und können nur auf dieser einen Seite angewendet werden. Ganz im Gegensatz zu einer *Browser Action*, welche im gesamten Browser in der Taskleiste lokalisiert ist und auf allen Webseiten angewendet wird. Mit der Hilfe von *Content Scripts* können allerdings Einschränkungen gemacht werden.

Für die Erweiterung wird eine *Browser Action* implementiert (wie in Programm 5.2 Zeile 17 deklariert), da die Extraktion von Interessen prinzipiell auf allen Webseiten durchgeführt werden soll (mehr dazu in Abschnitt 5.3.1).

Des weiteren beinhaltet die Erweiterung eine *Background Page*, welche im Hintergrund innerhalb desselben Kontexts des Browsers läuft. Die Logik wird in einer JavaScript-Datei über die Zeile 6 definiert. Prinzipiell gibt es hierfür zwei unterschiedliche Arten. Zum einen gibt es *Persistent Background*

[3] http://developer.chrome.com/extensions/index.html

Programm 5.2: manifest.json: Die Konfigurationsdatei der Google Chrome Erweiterung, welche die Eigenschaften der Erweiterung beinhaltet.

```
1  {
2  "name": "SemantLink",
3  "version": "1.0",
4  "manifest_version": 2,
5  "background": {
6     "scripts": ["jquery.js", "background.js"],
7     "persistent": false
8  },
9  "content_scripts": [
10   {
11     "matches": ["<all_urls>"],
12     "js": ["jquery.js", "contentscript.js"],
13     "all_frames": false,
14     "exclude_globs" : ["http://www.google.*", "*://www.facebook.*""]
15   }],
16 "manifest_version": 2,
17 "browser_action": {
18   "default_icon": "icon128.png",
19   "default_popup": "popup.html"
20 },
21 "options_page": "options.html",
22 "permissions": ["tabs", "notifications", "<all_urls>", "storage"]
23 }
```

Pages, welche immer im Hintergrund geöffnet und für die Kommunikation bereit sind. Zum anderen gibt es die *Event Pages*, welche nur dann geöffnet sind, wenn diese benötigt werden. Der Wechsel des Status innerhalb von *Event Pages* können zB mit der Registrierung von Events, oder dem Senden von Nachrichten, geregelt werden. Wie in der Konfigurationsdatei in Zeile 5 definiert, wird die Variante der *Event Page* verwendet, da diese nur dann als Prozess aktiv laufen, wenn die Funktionalität benötigt wird und ist somit schonend für die Performanz der Anwendung und auch des Browsers[4]. Zum Statuswechsel dient ein Event, welches bei einem geöffneten Tab ausgelöst wird. Sobald das Laden der Webseite abgeschlossen ist, wird die URL innerhalb des *background.js* über einen Ajax-Befehl an den Controller weitergereicht.

Prinzipiell kann zwischen der Erweiterung und einer geladenen Webseite kommuniziert werden. Zugriffe auf die URLs sind ohne zusätzlichem Aufwand möglich. Allerdings bieten Google Chrome Erweiterungen *Content Scripts* an, welche nur unter speziellen Bedingungen geladen werden. Dadurch können einzelne Webseiten von der Extraktion der Interessen ausge-

[4] http://developer.chrome.com/extensions/event_pages.html, zuletzt aufgerufen am 18.08.2013

Programm 5.3: Ein Event Listener welcher beim Erstellen eines Tabs im Browser reagiert.

```
1 chrome.tabs.onUpdated.addListener(function(tabId, changeInfo, tab) {
2   var url = tab.url;
3   var status = tab.status;
4 });
```

schlossen werden. Deswegen wird die Kommunikation über *Content Scripts* mit dem Senden von Nachrichten umgesetzt, siehe Programm 5.4.

Content Script

Ein *Content Script* läuft innerhalb des selben Kontexts der gerade aktiven Webseite und ermöglicht einen Zugriff auf Informationen wie zB Inhalt, Links, uvm. Innerhalb des Scripts wird die aktuell besuchte URL der Webseite entnommen und über das Senden von Nachrichten dem *Background Script* übermittelt, wie dem Programm 5.4 zu entnehmen ist. Prinzipiell wäre es auch ohne *Content Scripts* möglich, auf die aktuelle URL Zugriff zu erhalten, allerdings bieten *Content Scripts* zusätzlich die Möglichkeit, Ausnahmen für die Ausführung des Scripts zu definieren (wie in Zeile 14 mit der Eigenschaft `exclude_globs`). Ohne jener Eigenschaft würde es bei jeder Webseite aufgerufen werden, da es sich um eine *Browser Action* handelt. Da für diese Anwendung allerdings die Möglichkeit besitzen soll, dass Webseiten explizit aus der Analyse ausgeschlossen werden sollen können, wie zB Facebook, wird ein *Content Scripts* integriert, welche das Senden von Nachrichten mit der aktuell besuchten URL initiiert, siehe Programm 5.4.

Um im *Background Script* die Nachricht entgegennehmen zu können, wird ein Listener registriert, welcher eingehende Nachrichten abfängt und diese Information dem Controller über einen Ajax-Befehl weiterreicht.

UI Elemente

Die Erweiterung soll für den Benutzer Einstellungsmöglichkeiten bieten, so wie das Aus- und Einschalten der Datenermittlung, aber auch eine Anmeldung, um die Nutzung der Anwendung für mehrere Benutzer zu ermöglichen. Dafür werden User Interface Elemente mit eignen HTML Seiten konfiguriert (wie in Zeile 19 und 21 im Programm 5.2). Hierbei sind zwei zu unterscheiden:

options_page: beinhaltet ein Formular für die Anmeldung des Benutzers.

default_popup: beinhaltet die Möglichkeit, die Datenermittlung zu aktivieren bzw. zu deaktivieren und eine Verlinkung zu den Einstellungen (*options_page*).

Programm 5.4: Senden von Nachrichten.

```
1 //contentscript.js
2 chrome.runtime.sendMessage({url: document.URL});
3
4 //background.js
5 chrome.runtime.onMessage.addListener(function(request, sender,
      sendResponse) {
6   sendRequestToServer(request.url);
7 });
```

Um die Benutzerinformationen zwischen Server und Client auszutauschen, wird auf der Client-Seite auf die HTML5 localStorage Funktionalität zurückgegriffen. Wenn sich ein Benutzer anmeldet, wird dieser Benutzername darin gespeichert und per Ajax-Befehl übergeben.

5.3.2 Extraktion der Schlüsselwörter

Um von einer besuchten Webseite relevante Informationen für das Benutzerprofil zu extrahieren, werden bereits bestehenden APIs verwendet: Zemanta, AlchemyAPI und DBpedia Spotlight.

Herangehensweise

Zemanta bietet eine REST-Schnittstelle für die Verwendung der API an [50]:

`http://api.zemanta.com/services/rest/0.0/`

Für die Verwendung von Zemanta wird eine eigene Klasse mit dem Senden des HTTP-Requests und dem Parsen der Antwort erstellt, damit ein einfacher Aufruf im Controller möglich ist und gegebenenfalls auch ein schneller Austausch der APIs ermöglicht ist.

Als Methode von Zemanta wird `zemanta.suggest` verwendet, um Schlüsselwörter und Entitäten abzufragen [49]. Als Parameter ist entweder HTML- oder Klartext erlaubt. Die Übergabe von kompletten HTML-Webseiten bzw. URLs wird nicht empfohlen, weshalb vom Controller zunächst der Klartext aus der URL extrahiert werden muss. Hierfür wird die Alchemy API verwendet (*Text Extraction*[5]), welche über ein SDK in das Projekt integriert wird, das alle nötigen Schritte abdeckt.

Wie in Abschnitt 4.6 bereits erwähnt, werden von Zemanta acht relevante Wörter oder Phrasen als Schlüsselwörter zurückgegeben. Ein jedes Schlüsselwort beinhaltet ebenfalls eine Relevanz zwischen 0 und 1, welche sich auf den Inhalt des Textes bezieht. Basierend auf den Ergebnissen der Master Thesis

[5] http://www.alchemyapi.com/api/text-extraction/

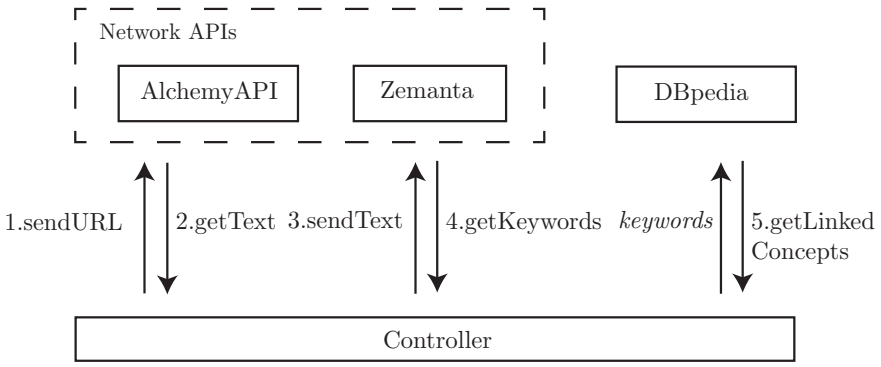

Abbildung 5.3: Ablauf bei der Extraktion der Schlüsselwörter.

von Bauer [2], liefert Zemanta viele Schlüsselwörter mit 0.05 Relevanz zurück. Dieser Wert zeigt allerdings eine unsichere Zuordnung des Begriffs zum Text, weshalb jene für eine Weiterverarbeitung nicht berücksichtigt werden sollten. Somit werden nur jene Schlüsselwörter extrahiert, die eine Relevanz von > 0.1 liefern. Dieser Wert ergab bei den Tests stabile und relevante Ergebnisse.

5.3.3 Extraktion der DBpedia Kategorien

Die extrahierten Schlüsselwörter von Zemanta dienen nun als Basisdaten für deren weitere Anreicherung. In Abbildung 5.3 ist der gesamte Ablauf der Extraktion abgebildet.

Um die passende DBpedia Entität zu finden, wird die DBpedia Spotlight API [6], welche als Webservice über folgende REST-Schnittstelle aufgerufen wird:

```
http://spotlight.sztaki.hu:2222/rest/annotate
```

Über POST werden die Parameter `text`, `confidence=0.1`, `format=rdfxml` übergeben und als Antwort wird die gefundene Entität aus DBpedia mit dessen eindeutigen URI zurückgeliefert. Diese URI dient in weiterer Folge als Brücke zu den weiteren Informationen innerhalb von DBpedia. Die verlinkten Kategorien von einer Entität werden mit Attribut `dcterm:subject` erhalten. Um von den einzelnen Kategorien wiederum zu weitere Kategorien zu gelangen, wird das Attribut `skos:broader` verwendet. Diese Zwischenkategorien bis hin zu einer passenden Hauptkategorie wird mit folgendem Suchalgorithmus ermittelt.

[6] https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki

Suchalgorithmus

Ein Begriff bzw. Kategorie kann mehrere Überkategorien haben. Deswegen muss auf jeder Ebene entschieden werden, welche Überkategorie am besten mit dem Ursprungsbegriff übereinstimmt und welcher Weg hin zur Lösungskategorie verfolgt werden soll. Ein Suchalgorithmus ist zuständig, dies zu einem frühen Zeitpunkt zu entschieden. Als Heuristik dafür wird auf die Ähnlichkeit zwischen zwei Entitäten zurückgegriffen. Dabei wird bei jedem Schritt überprüft, welche Kategorie auf einer Ebene die meisten Ähnlichkeiten mit dem Ursprungsbegriff besitzt. Der Algorithmus ist im Pseudocode 5.1 beschrieben.

Ähnlichkeitsalgorithmus

Als Grundlage für die Berechnung der Ähnlichkeit zwischen zwei Entitäten werden dessen verbundenen Kategorien bis zu drei Ebenen in iterativer Weise verwendet. Wenn es eine große Menge an Überschneidungen von den verglichenen Kategorien gibt, dann ist die Ähnlichkeit größer, als wenn weniger gleiche Überschneidungen vorkommen. Die Berechnung basiert auf die *Normalized Google Distance* [33]

$$NGD(x,y) = \frac{\max[\log f(x), \log f(y)] - \log f(x,y)}{\log N - \min[\log f(x), \log f(y)]}, \tag{5.1}$$

wobei $f(x)$ die Anzahl der Treffer für den Ursprungsbegriff x, $f(y)$ die aktuelle Katgeorie y und N die gesamten Artikel aus DBpedia benennt. Der angewendete Algorithmus zum Finden des Lösungsgraphen wird in Pseudocode 5.1 beschrieben.

Die Iteration durch den DBpedia-Graphen durchläuft solange, bis eine Lösungskategorie aus der *Main topic classification* gefunden wird. Wird eine passende Kategorie mit der größten Ähnlichkeit zum Ausgangsbegriff gefunden, kann dieser in den Lösungspfad integriert werden. Dabei muss darauf geachtet werden, dass an der richtigen Stelle eingefügt wird. Es kann sein, dass ein verfolgter Weg abgebrochen wird und an einer neuen Stelle im Kategoriebaum von DBpedia weiter verläuft, da dieser Weg als besser empfunden wird. Somit müssen die Kategorien dazwischen aus dem bereits gespeicherten Lösungspfad entfernt und das neue Element an der richtigen Stelle mit dem richtigen Vorgänger integriert werden.

5.4 Speicherung in das Benutzerprofil

Für die Speicherung der Daten in RDF-Form wird auf bestehende Vokabulare zurückgegriffen: Dublin Core, FOAF, SKOS, welche bereits in Abschnitt 4.5.1 beschrieben werden.

Algorithmus 5.1: Finden des besten Pfades im Kategoriebaum von DBpedia.

```
iterativeSearchForCategories(keyword)
    node ← keyword
    foundMainTopic ← false
    bestCategoryPath add keyword
    mainTopics ← getMainTopicsOfDBpedia
    level ← 0
    while ( !foundMainTopic ) do
        children ← getCategories(node)
        for i = 0 to sizeOf(mainTopics) do
            if (children contains mainTopics_i) then
                foundMainTopic ← true
                addToPathAtLevel(mainTopics_i, level)  ▷ Einfügelevel
aktualisieren
            end if
        end for
        if !foundMainTopic then
            bestNode ← getBestNode(children)
            if !bestNode then
                children ← openNodes
                bestNode ← getBestNode(children)
                level ← bestNode.level
            end if
            openNodes add bestNode
            node ← bestNode
        end if
    end while
    return bestCategoryPath
end
```

Im folgenden Abschnitt wird auf die Implementierung der Speicherung eingegangen, basierend auf dem Benutzermodell in Abbildung 4.5.

5.4.1 Speicherung der Benutzerdaten

Der Benutzer wird mit dem Typ von `foaf:Person` gespeichert und mit weiteren Attributen, wie `foaf:name` und `foaf:username` ergänzt. In Turtle-Form schaut das *INSERT*-Statement wie folg aus:

Programm 5.5: Speicherung der Benutzerinformation.

```
1 ex:MaxMustermann rdf:about      foaf:Person ;
2   foaf:name      'Max Mustermann' ;
3   foaf:username  'MaxMustermann' .
```

Programm 5.6: Speicherung der besuchten Webseite.

```
1 <http://mashable.com/2013/09/06/compress-iphone-photos/>
2   rdf:type       foaf:Document ;
3   dcterm:date        '1376573424'^^xsd:int ;
4   ex:visited     'false'^^xsd:boolean ;
5   ex:visitedBy   ex:MaxMustermann .
6
7 ex:MaxMustermann  foaf:interest
8   <http://mashable.com/2013/09/06/compress-iphone-photos/>  .
```

Speicherung der Interessen

Um die Interessen des Benutzers zu speichern, werden sowohl die besuchte Webseite, als auch die ermittelten Konzepte inklusive einer Verbindung zum Benutzer und wieder zurück gespeichert.

Eine vom Benutzer betrachtete Webseite wird als Typ `foaf:Document` mit dessem Erstelldatum `dcterm:date` und Titel `dcterm:title` ausgezeichnet. Um die besuchten Webseiten auch als solche abzuspeichern, wird jede mit dem Attribut `ex:visited` als boolescher Wert und `ex:visitedBy` mit der URI des Benutzers belegt. Somit können Anwendungen dies als Filterbedingung inkludieren. Die Verbindung zwischen der Webseite und dem Benutzer wird mit dem Attribut vom FOAF-Projekt `foaf:interest` belegt.

Die extrahierten Konzepte einer besuchten Webseite werden ins Benutzerprofil als Typ `skos:Concept` mit dessen Bezeichnung als `dcterm:title` belegt. Grundsätzlich wird unterschieden, ob ein Konzept eine Hauptkategorie (`skos:topConceptOf` bzw. `ex:isPrimaryTopic`), ein Schlüsselwort (`ex:isKategory`) oder ein Zwischenkonzept ist. Diese Informationen werden in dieser Form in das Profil mit dessen Beziehung zueinander gespeichert. Ein Schlüsselwort steht mit all dessen verbundenen Kategorien über `dcterm:subject` in Verbindung, genauso wie die extrahierten Konzepte von dessen Webseite.

5.4.2 Auswahl der relevanten Interessen

Nach diesem Vorgehen befinden sich im Profil viele Begriffe, sowohl spezifische Schlüsselwörter, als auch allgemeinere Konzepte bis hin zu den Hauptkategorien mit unterschiedlichen Verknüpfungen zueinander. Dies ermöglicht

Programm 5.7: Speicherung der extrahierten Themengebiete.

```
...
//Speicherung der URL mit der Verbindung zum extrahierten Begriff
<http://mashable.com/2013/09/06/compress-iphone-photos/>
      dcterm:subject  db:IPhone .

//Speicherung eines Zwischenkonzepts
db:IPhone   rdf:type            skos:Concept ;
            dcterm:title        'IPhone' ;
            dcterm:subject      db:Category:Technology ;
            skos:hasTopConcept  db:Category:Technology ;
            ex:connectionWeight '0.8765'^^xsd:boolean ;
            ex:isKeyword        'true'^^xsd:boolean ;
            ex:count            '5'^^xsd:int .

//Speicherung der Hauptkategorie
db:Category:Technology  rdf:type  skos:Concept ;
                        dcterm:title  'Technology' ;
                        ex:isPrimaryTopic  'true'^^xsd:boolean ;
                        skos:topConceptOf  db:Category:Technology .

<http://mashable.com/2013/09/06/compress-iphone-photos/>
      foaf:primaryTopic db:Category:Technology ;
      foaf:topic        db:IPhone .

db:Category:Technology  foaf:isPrimaryTopicOf
      <http://mashable.com/2013/09/06/compress-iphone-photos/> ;
      skos:topConceptOf      db:IPhone .
...
```

reichhaltige Abfragen für unterschiedliche Anwendungen.

Ein Begriff besitzt noch eine Gewichtung, welche abhängig ist von der Ähnlichkeit zum Ursprungsbegriff, aber auch einem Zählerattribut `ex:count`, welcher bei erneutem Aufkommen des Konzepts erhöht wird. Mit diesen Informationen werden die Begriffe im Benutzerprofil angezeigt.

1. Ermittlung der Top 10 Schlüsselwörter, die direkt von der API von den besuchten Webseiten extrahiert werden (Programm 5.8).
2. Erweiterung mit den Top 3 Zwischenkategorien aus DBpedia jeweils von den Top 10 Schlüsselwörtern, welche die stärkste Ähnlichkeit mit dem Ausgangsbegriff vorweisen (Programm 5.9).
3. Erweiterung mit den Top 3 Hauptkategorien (Programm 5.10).

Programm 5.8: SPARQL-Select-Statement für die Ermittlung der Top 10 Schlüsselwörter.

```
SELECT * WHERE
{
?keyword   rdf:type skos:Concept ;
           dcterm:title ?name ;
           ex:count ?count ;
           ex:isKeyword ?keywordBool .
<http://xmlns.com/foaf/0.1/MaxMustermann> foaf:topic_interest ?keyword .
FILTER ( xsd:boolean(?keywordBool) )
}
GROUP BY ?keyword
ORDER BY DESC(xsd:integer(?count))
LIMIT 10';

```

Programm 5.9: SPARQL-Select-Statement für die Ermittlung der Top 3 Zwischenkonzepten.

```
SELECT * WHERE
{
?keyword  rdf:type skos:Concept ;
          dcterm:title ?keywordName ;
          dcterm:subject ?concept ;
          ex:isKeyword ?keywordBool  .
?concept  ex:count ?count ;
          dcterm:title ?name ;
          ex:connectionWeight ?weight .
FILTER ( xsd:boolean(?keywordBool) )
FILTER ( ?keywordName = "KEYWORDNAME_PLACEHOLDER")
<http://xmlns.com/foaf/0.1/MaxMustermann> foaf:topic_interest ?keyword .
<http://xmlns.com/foaf/0.1/MaxMustermann> foaf:topic_interest ?concept .
}
GROUP BY ?concept
ORDER BY DESC(xsd:integer(?weight) )
LIMIT 3';

```

5.5 Zusammenfassung

Die Interessen des Benutzers werden in einem Benutzerprofil gespeichert. Als Interessen dienen extrahierte Schlüsselwörter von besuchten Webseiten, welche einer weiteren Analyse unterzogen werden. Diese Analyse basiert auf die Artikel von Wikipedia, die als Entitäten in der DBpedia Datenbank über eine eindeutige URI verfügbar sind. Mit Hilfe der verbundenen Kategorien werden Zwischenkonzepte bis hin zu einer gefundenen passenden Hauptka-

Programm 5.10: SPARQL-Select-Statement für die Ermittlung der Top 3 Hauptkonzepten.

```
SELECT * WHERE
{
?concept  rdf:type skos:Concept ;
          dcterm:title ?name ;
          foaf:isPrimaryTopicOf ?url ;
          ex:count ?count .
<http://xmlns.com/foaf/0.1/MaxMustermann> foaf:topic_interest ?concept .
}
GROUP BY ?concept
ORDER BY DESC(xsd:integer(?count))
LIMIT 3'

```

tegorie gesucht. Die Zwischenkonzepte dienen als erweiterte Informationen zum extrahierten Begriff, die zum einen das Schlüsselwort näher beschreiben, und zum anderen alternative Begriffe liefern. Die Hauptkategorien erfüllen den Zweck, dass allgemeinere Themengebiete im Profil abgedeckt sind. Mit diesen zwei Erweiterungen wird das Wissen des Benutzers angereichert und Anwendungen können sowohl spezifische, als auch allgemeine Interessen in die Filterung von Informationen integrieren.

Kapitel 6

Evaluierung

In diesem Kapitel soll gezeigt werden, welche Auswirkung es hat, wenn ein Benutzerprofil mit extrahierten Schlüsselwörter und automatisierten Zusatzinformationen in Form von Kategorien aus DBpedia aufgebaut wird. Außerdem soll überprüft werden, ob die definierten Ziele erreicht wurden.

6.1 Analyse der Ergebnisse

Folgende Abschnitte analysieren die einzelnen Prozessschritte, die für den Aufbau des Benutzerprofils implementiert wurden.

6.1.1 Datenerfassung

Die Google Chrome Erweiterung läuft stabil innerhalb des Browsers. Leistungseinbuße des Browsers konnten nicht beobachtet werden. Da die Variante von *Event Pages* verwendet wurde, wird ein Prozess für die Erweiterung nur dann gestartet, wenn eine neue Webseite geöffnet wird und zwar solange, bis die Benutzerinformationen übermittelt ist. Die leichte Handhabung der Erweiterung ist ebenfalls gegeben, indem diese schnell über das in Abbildung 6.1 dargestellte Popup aktiviert bzw. deaktiviert werden kann, um die Privatsphäre des Benutzers zu berücksichtigen. Wenn die Erweiterung deaktiviert ist, wird keine Information austauscht.

Die automatische Extraktion von relevanten Begriffen basierend auf unstrukturierten Texten von Webseiten wurde in der Literatur schon mehrmals analysiert. Die gewählte Variante mit Zemanta führte in der englischen Sprache zu einem gutem Ergebnis. Es wurden nicht nur Begriffe aus den Texten extrahiert, sondern auch mit der Bedeutung des Textes verbundene Konzepte gefunden. Die APIs Zemanta und Alchemy waren im gesamten Entwicklungsprozess stabil und konnten ohne Ausfälle verwendet werden. Ganz im Gegensatz zur lokalen DBpedia Datenbank, welche oftmals nicht erreichbar war.

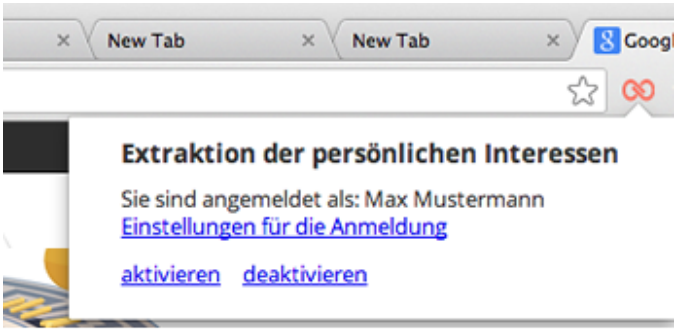

Abbildung 6.1: Die Google Chrome Erweiterung innerhab der Taskleiste positioniert für einen schnellen Zugriff.

Bei anderen Sprachen gab es noch gröbere Probleme bei der Zemanta API. Diese lieferten zu Beginn selten zufriedenstellende Ergebnisse in der deutschen Sprache, welche von Zemanta zur Zeit nicht unterstützt wird. Teilweise wurden trotzdem brauchbare Ergebnisse geliefert. Die Relevanz der extrahierten Schlüsselwörter befand sich meist unterhalb von 0.01, weshalb diese Analyse nicht aussagekräftig ist und sich hier rein auf englische Begriffe beschränkt wurde. Ein großes Problem stellen noch die dafür notwendigen Requests an das Webservice dar, wodurch lange Wartezeiten aufgetreten sind. Des weiteren gibt es eine Beschränkung auf eine tägliche Anzahl an Requests, welche bei 1.000 liegt[1]. Dieses Problem betrifft die AlchemyAPI ebenfalls, wobei die Begrenzung hier bei 30.000 liegt[2].

6.1.2 Analyse des Benutzerprofils

Um zu analysieren, ob der verwendete Ansatz Potential hat, wird ein Ausschnitt aus einem Testprofil gezeigt. Dieser Ausschnitt wird mit zehn Schlüsselwörtern limitiert und ist in Abbildung 6.2 zu sehen. Auf der linken Seite sind lediglich die direkt von der API extrahierten Schlüsselwörtern aufgelistet. Auf der rechten Seite befindet sich das Profil inklusive DBpedia Konzepten nach den beschriebenen Kriterien aus Abschnitt 5.4.2, die hier aufgelistet sind:

1. Ermittlung der Top 10 Schlüsselwörter, die direkt von der API von den besuchten Webseiten extrahiert wurden (Programm 5.8).
2. Erweiterung mit den Top 3 Zwischenkategorien aus DBpedia jeweils von den Top 10 Schlüsselwörtern, welche die stärkste Ähnlichkeit mit dem Ausgangsbegriff vorweisen (Programm 5.9).
3. Erweiterung mit den Top 3 Hauptkategorien (Programm 5.10).

[1] http://developer.zemanta.com/docs/
[2] http://www.alchemyapi.com/api/calling-the-api/

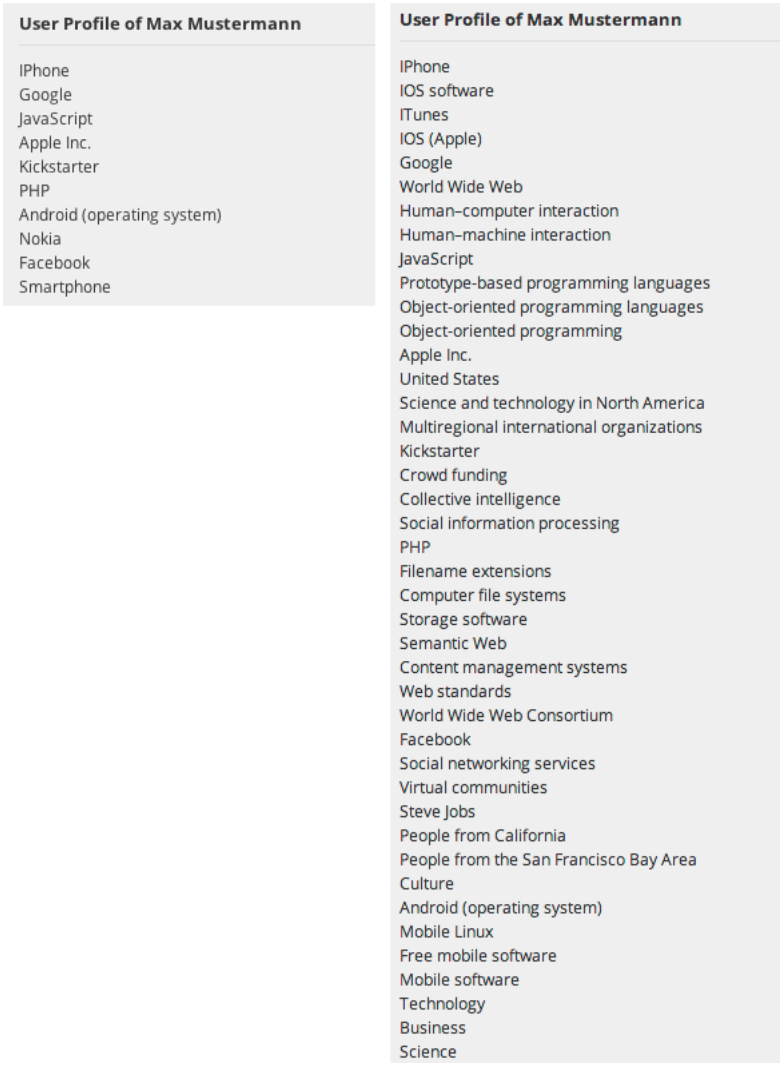

Abbildung 6.2: Auszug aus einem Testprofil. Auf der linken Seite das Profil mit den extrahierten Schlüsselwörtern von der API. Auf der rechten Seite das Profil inklusive der analysierten Konzepten aus DBpedia.

Das Profil wurde mit den Zwischenkonzepten und Hauptkategorien angereichert. Dies soll dazu führen, dass das Benutzerprofil erweitert wird und somit ein breites Wissen über den Benutzer vorhanden ist. Die Zwischenkonzepte sollen für erweiterte Begriffe zu den Schlüsselwörtern dienen, um weitere verwandte ähnliche Begriffe zu inkludieren. Die Hauptkategorien sollen die allgemeinen Interessen des Benutzers abdecken. Um zu analysieren, ob diese Anreicherung brauchbare Resultate liefert, werden einzelne Schlüsselwörter aus dem generierten Profil ausgewählt und betrachtet.

In folgender Tabelle 6.1 werden die Begriffe im Profil aufgelistet und zu den Schlüsselwörtern die jeweiligen Anreicherungen von DBpedia Kategorien aufgelistet:

Tabelle 6.1: Anreicherung der Schlüsselwörter mit DBpedia Kategorien innerhalb des Profil.

Schlüsselwort	**Anreicherung**
IPhone	{IO Software, ITunes, IOS (Apple), Technology}
Google	{Word Wide Web, Human-computer interaction, Human-machine interaction}
JavaScript	{Prototype-based programming language, Object-oriented programming language, Object-oriented programming}
Facebook	{Social networking services, Virtual communities, Collective Intelligence}
Apple Inc.	{United States, Multiregional international organisations, Science and technology in North America}
Android (operating system)	{Mobile Linux, Mobile software, Free mobile software}
Kickstarter	{Crowd funding, Collective Intelligence, Social information processing}
PHP	{Filename extensions, Computer file systems, Storage software}
Semantic Web	{Content management systems, Web Standards, Word Wide Web Consortium}
Steve Jobs	{People from California, People from the San Francisco Bay Area, Culture}

Schlussfolgerungen

Die oben beschriebenen Beispiele liefern mehr Informationen zu dem Ursprungsbegriff und führen weitere verwandte Begriffe an. Zum Beispiel ist anzunehmen, das für diesen ebenfalls *IOS Software* und auch das dazugehörige Betriebssystem *IOS (Apple)* von Interesse sein kann. Auch der Begriff *ITunes* ist ebenfalls verwandt mit *IPhone* und erweitert hiermit das Wissen über den Benutzer. Für einen Computer wäre diese Interpretation nicht möglich, ohne auf semantische Hilfsmittel und eine aussagekräftige Verknüpfung von Begriffen zurückgreifen zu können.

Die Hauptkategorien *Technology*, *Business* und *Science* decken zusätzlich zu den eher spezifischeren Themengebieten aus Schlüsselwörtern und Zwischenkonzepten allgemeinen Interessensbereich des Benutzers ab. Da ver-

mehrt die zuvor genannten Kategorien auf Grund von der Analyse ermittelt wurden, decken diese drei Begriffe die allgemeinen Interessensbereiche des Benutzers ab. Insgesamt sind 24 Hauptkategorien von DBpedia bereitgestellt. Bei dieser Auswahl ist wichtig, dass nicht zu viele Hauptkategorien in das Profil miteinfließen, da ansonsten die Bestimmung von relevanten und nicht relevanten Informationen zu ungenau wird.

Bei dem Begriff *JavaScript*, *Facebook*, *Android (operating system)*, *Kickstarter* und *Semantic Web* werden ebenfalls zufriedenstellende Ergebnisse geliefert. Die Begriffe befinden sich semantisch gesehen im selben Kontext des Schlüsselworts und stellen verwandte Begriffe dar.

Einige Anreicherungen zeigen optimierungsbedarf, denn ein Mensch würde den Begriff *Google* nicht unmittelbar mit *Human-computer interaction* oder *Human-machine interaction* beschreiben. Hier wären Begriffe, die mit dem Begriff *Search engines* näher verwandt sind, zu erwarten.

Der Begriff *Apple Inc.* hingegen wird nicht zufriedenstellend angereichert. Hier ist das Problem, dass die zugewiesenen Zwischenkonzepte zu allgemein sind. Dies liegt daran, dass die direkten Kategorien dieses Wikipedia Artikels bereits sehr allgemeine Themenbereiche zugewiesen hat[3]:

- *category:Computer_ companies_ of_ the_ United_ States*,
- *category:Computer_ hardware_ companies*,
- *category:Retail_ companies_ of_ the_ United_ States*,
- uvm.

Auch beim Begriff *Steve Jobs* wäre wünschenswert, dass Begriffe wie *Apple* gefunden werden. Da allerdings bei DBpedia neben spezifischen wie *Apple_ Inc.* vermehrt personenbezogene allgemeine Wikipedia-Kategorien zugewiesen sind, werden diese anhand der Ähnlichhkeitsberechnung auf Grund von der Übereinstimmung der zugeordneten Kategorien als besser interpretiert. Hierfür müsste eine Blacklist an Kategorien angelegt werden, welche spezielle Arten von Kategorien ausschließt.

Die verknüpften Kategorien bieten sich somit bei diesen beiden Begriffen nicht als optimale Quelle für die Anreicherung an. Als alternativer Ansatz könnten anstelle von den Kategorien verlinkte Ressourcen für die nähere Beschreibung der Entität dienen. Dieser Ansatz wurde bereits in der Literatur getestet und müsste dem hier verwendeten in einer weiteren Testphase gegenübergestellt werden. Bei den anderen in der Tabelle beschriebenen Begriffen ist die Anreicherung durchaus zufriedenstellend und zeigt, dass eine automatisierte Anreicherung von Begriffen mit DBpedia Kategorien für Benutzerprofile brauchbare Ergebnisse liefert.

[3] http://dbpedia.org/page/Apple_Inc. zu finden unter *dcterms:subject*, zuletzt aufgerufen am 15.09.2013

6.1.3 Fazit

Die Evaluierung dieser automatisierten angereicherten Begriffe mit der Hilfe von DBpedia Kategorien zeigt, dass der Ansatz brauchbare Ergebnisse liefert, um die Interessen des Benutzers zu erweitern. Der Ansatz ist allerdings noch nicht ideal. Bei der Anreicherung von manchen Begriffen werden keine zufriedenstellende Resultate geliefert, die als Interessenerweiterung dienen. Außerdem ist die Bearbeitungszeit bei vielen zugewiesenen Kategorien sehr lange. Hierfür müsste eine Optimierung durchgeführt werden, sodass bereits auf der ersten Kategorieebene eine Vorauswahl an brauchbaren Kategorien durchgeführt wird.

Besonders bei anderen Sprachen ist dieser Ansatz noch sehr fehleranfällig, da die meisten APIs für die englische Sprache optimiert sind. Der Grund dafür ist, dass diese hauptsächlich englische Datenquellen als Basis für die Analyse verwenden. Allerdings wird diesem Problem versucht entgegenzutreten, indem nur Konzepte im Profil ausgegeben werden, die oftmals extrahiert und analysiert werden. Zum anderen ist der Ähnlichkeitsalgorithmus in manchen Fällen nicht zuverlässig. Im DBpedia Graph wird ein falscher Weg gewählt und speichert in das Profil Konzepte, die nach menschlicher Bewertung eine geringe Ähnlichkeit aussagen. Für eine Maschine ist die Bestimmung der Ähnlichkeit zwischen zwei Begriffen nicht einfach und muss in diesem Ansatz noch optimiert werden.

6.2 Erreichte und nicht erreichte Ziele

In der Spezifikation wurde als Anforderung definiert, dass das System ohne explizite Handlungen des Benutzers dessen Interessen ermitteln soll, ohne diesen abzulenken. Dies wurde mit der Erweiterung innerhalb des Browsers erreicht. Die einzige Tätigkeit, die vom Benutzer verlangt wird, ist die Installation der Erweiterung und eine Zustimmung für den Zugriff auf die besuchten Webseiten bzw. eine Deaktivierung, falls diese gewünscht wird. Außerdem wurde spezifiziert, dass alle besuchten Webseiten analysiert werden können, wenn der Benutzer die Erweiterung aktiviert hat. In der Browser Erweiterung wurde die Variante einer *Browser Action* gewählt, welche auf alle möglichen Webseiten den Zugriff ermöglicht.

Die Browser Erweiterung muss in weiterführenden Tests beobachtet werden, ob diese Leistungseinbuße des Browsers verursacht. Mit der Umsetzung von *Event Pages* anstelle von *Persistent Background Pages*, wird versucht dieses Problem zu minimieren.

Mit der Login-Möglichkeit gibt es keine Beschränkung für die Nutzung des Systems auf nur einen Benutzer bzw. Computer. Die Erweiterung kann an unterschiedlichen Standorten im Browser Google Chrome installiert werden und ermöglicht einen Zugriff auf das eigene Profil. Da eine Google Chrome Browser Erweiterung implementiert wurde, ist das System an den Browser

Google Chrome gebunden.

Semantische Technologien haben für den Aufbau des Benutzerprofiles geholfen, indem ein RDF-Triple-Store als Datenbankform gewählt wurde. Bereits bestehende Konzepte wie FOAF, Dublin Core und SKOS wurden eingesetzt, um die Daten mit anderen in Verbindung zu bringen, da alle neuen Daten mit einer URI ausgezeichnet sind. Die Daten sind allerdings noch nicht öffentlich für andere zugänglich, da dafür noch Optimierungen notwendig sind. Es wurden allerdings die Daten bereits nach den Linked Data-Prinzipien aufgebaut. Mit einer Linked Data View könnten die Daten zugänglich gemacht werden[4].

Die Auswahl der relevanten Interessen liefern zufriedenstellende Begriffe zu den extrahierten Schlüsselwörter. Die Anreicherung beschreibt die extrahierten Begriffe näher und erweitert das Wissen über den Benutzer. Um die Evaluierung des ermittelten Profils optimal durchführen zu können und die Auswirkung qualitativ testen zu können, müsste ein Anwendungsszenario die gesammelten Interessen integrieren und für die Personalisierung anwenden.

[4]Nähere Informationen für die Veröffentlichung von Daten als Linked Data sind folgender Webseite zu entnehmen. http://wifo5-03.informatik.uni-mannheim.de/bizer/pub/LinkedDataTutorial/, zuletzt aufgerufen am 18.09.2013

Kapitel 7

Resümee

7.1 Zusammenfassung

Der Aufbau von Benutzerprofilen mit implizit automatisch generierten Interessen wurde in diesem Ansatz behandelt. Die Erfassung der Benutzerinformationen wurde mit einer Google Chrome Browser Erweiterung durchgeführt. Die Anreicherung von extrahierten Schlüsselwörtern mit weiteren Konzepten von Wikipedia ermöglicht, dass die Informationen über den Benutzer erweitert werden. Verwandte Themenbereiche und allgemein ermittelte Hauptkategorien werden auf Grund von den verknüpften Kategorien von DBpedia automatisiert für die Deckung von allgemeinen Interessengebieten ermittelt. Diese Informationen dienen als Basis für die Ermittlung der Interessen des Benutzers und wurden in eine hierarchische Struktur innerhalb eines Benutzerprofils abgebildet. Semantische Technologien dienen als Unterstützung für die Speicherung und Analyse von verwandten Interessen. Die Gewichtung der Interessen ist abhängig von der Ähnlichkeit zwischen den extrahierten Schlüsselwörtern und den ermittelten Konzepten aus Wikipedia und der Häufigkeit dessen Vorkommens. Die am öftesten auftretenden Schlüsselwörter im Profil werden als die aktuellen Interessen interpretiert und mit dessen verknüpften Konzepten im Profil angezeigt. Diese Interessen können von Filter- und Personalisierungssystemen als Basis verwendet werden, um die Informationsbeschaffung an den Benutzer anzupassen. Die gewonnen Ergebnisse sollen als Basis für weiterführende Arbeiten dienen.

7.2 Fazit

Diese Ausarbeitung ermöglichte es, einen tiefen Einblick zu bekommen, wie mit Hilfe von automatisierten semantischen Mechanismen ein hierarchisches Benutzerprofil von Wikipedia Konzepten als Darstellung für die Interessen aufgebaut werden kann. Die Evaluierung des Testprofils ist zufriedenstellend, allerdings in keinster Weise optimal und noch ausbaufähig. In manchen Be-

reichen benötigt dieser Ansatz noch Optimierungen, wie zum Beispiel bei der Bestimmung von ähnlichen, relevanten bzw. verwandten Begriffen, eine eigene Textextraktion zur Entlastung der Netzwerkzugriffe und eine Anpassung bei der Gewichtung der Interessen im Profil. Hierfür sind vor allem langzeitige Testphasen nötig, um die Auswirkung bei großen Profilen zu beobachten. Für eine kommerzielle Nutzung ist der Ansatz somit noch nicht geeignet.

7.3 Ausblick

Um eine ausführliche Evaluierung der Benutzerprofile zu ermöglichen, muss die Brauchbarkeit der Profile innerhalb eines Anwendungsszenaris getestet werden. Dafür würden sich zB personalisierte Suchanfragen oder Empfehlungen von Informationen in Form von personalisierten Newsfeeds eignen, womit überprüft werden kann, ob dieser Ansatz des Profilaufbaus eine Unterstützung für Personalisierungsvorgänge zeigt. Mit der Hilfe von einer Testphase über einen längeren Zeitraum (mind. 1-2 Monate) könnte die Überprüfung durchgeführt werden. Der lange Zeitraum ist deswegen notwendig, da das System Wissen über den Benutzer erwerben muss. Je mehr Informationen vorhanden sind, desto besser kann die Personalisierung durchgeführt werden.

Die Ermittlung der Interessen ist aktuell für die englische Sprache optimiert, da Zemanta prinzipiell keine anderen Sprachen unterstützt. DBpedia Spotlight unterstützt seit kurzem viele unterschiedliche Sprachen, unter anderem auch Deutsch, welche als Alternative in Betracht gezogen werden kann. Allerdings unterstützt DBpedia keine semantische Analyse bei der Datenermittlung, sondern führt lediglich eine Annotation mit DBpedia Entitäten durch. Die Weiterentwicklung der APIs führt zu ständiger Optimierung der Ergebnisse. Zum jetzigen Zeitpunkt gibt es bereits zahlreiche Funktionalitäten, die zu Beginn dieser Arbeit von Zemanta noch nicht angeboten wurden. Zum Beispiel beinhalten die Ergebnisse bereits einen Kategoriepfad aus der DMOZ Ontologie für die einzelnen extrahierten Schlüsselwörter, wie `Top/Science/Anomalies_and_Alternative_Science/Astronomy, _Alternative/Planetary_Anomalies`[1]. Zuvor wurde nur eine Hauptkategorie zurückgeliefert. Dies könnte in einer weiteren Ausführung getestet werden, ob diese Kategorien bessere Resultate für das Benutzerprofil liefern.

Da viele unterschiedliche Verknüpfungen zwischen den Begriffen gespeichert werden, allerdings nicht alle in der beschriebenen Variante für die Auswahl der relevanten Interessen eine Verwendung hatten, bietet dies einen Spielraum für weitere Variationen. Zum Beispiel könnten Anwendungen nur die Hauptkategorien des Benutzers entnehmen und somit eine Personalisierung basierend auf sehr allgemeinen Interessengebieten durchführen.

[1] Dieses Beispiel wurde von http://developer.zemanta.com/docs/suggest/ entnommen, zuletzt aufgerufen am 18.09.2013

Anhang A

Installationsanweisungen

A.1 Installation PHP-Anwendung

- Apache Server installieren und starten
- MySQL[1] installieren und starten
- Projektordner `source` in das Root-Verzeichnis des Servers legen (*htdocs-Ordner*) und ev. umbenennen
- Unter http://localhost/source/ die Anwendung aufgerufen und im Formular einen neuen Benutzer anlegen (zB ID = `Max_Mustermann`)
- Die Datenbankfelder für das Profil werden automatisch erstellt
- Sesame Triple Store Setup lokal installieren (siehe A.2)
- Google Chrome Browser Plugin installieren A.3

A.2 Installation DBpedia Datenbank (lokal)

1. Tomcat[2] herunterladen und installieren
2. Sesame SDK[3] herunterladen
3. Sesame SDK Dateien in das `%TOMCAT_HOME%`-Verzeichnis kopieren
4. DBpedia-NT-Dump von *Article Categories* und *Categories (SKOS)* herunterladen[4] und entpacken
5. Tomcat starten
6. Im Browser http://localhost:8080/openrdf-work-bench/ aufrufen
7. Neuen Triplestore unter *New Repository* anlegen mit `Parameter Type = Native Java Store` und eine beliebige `ID`

[1] http://dev.mysql.com/downloads/

[2] http://tomcat.apache.org/download-70.cgi

[3] http://www.openrdf.org/download_sesame2.jsp oder auf der CD unter `Projekt/Ressourcen`

[4] http://wiki.dbpedia.org/Downloads38 oder auf der CD unter `Projekt/Ressourcen`

8. DBpedia `.nt`-Datei als `RDF Data File` und mit dem Format `N-Triples` unter *Add* hochladen

A.3 Installation Google Chrome Browser Erweiterung

- Datei `Projekt/BrowserPlugin.zip` entpacken und überprüfen, ob die in der Datei `config.xml` angegebene baseURL für die Anwendung mit der lokalen Konfiguration überein stimmt.
- Google Chrome Browser herunterladen und installieren (http://www.google.com/intl/de/chrome/browser/)
- Google Chrome Browser öffnen
- Im Browser chrome://extensions öffnen
- *Entwicklermodes* aktivieren
- *Entpackte Erweiterung laden* anklicken
- Verzeichnis *Browser Plugin* aus Zip-File auswählen.
- Checkbox *Aktiviert* muss ausgewählt sein

Anhang B

Inhalt der CD-ROM

B.1 Masterarbeit (PDF)

Pfad: /

_DaBa.pdf . . . . . Aufbau von Benutzerprofilen für personalisierte Systeme mit Hilfe von Semantic Web Technologien

B.2 Projekt

Pfad: /Projekt

BrowserPlugin.zip . . . Google Chrome Erweiterung als Zip-Datei

source/*... . . . . PHP-Anwendung des Projekts

article_categories_en.nt.bz2 DBpedia-Dump für die Beziehung zwischen Artikel und Kategorie

skos_categories_en.nt.bz2 DBpedia-Dump für die Beziehung zu internen Kategorien

openrdf-sesame-2.7.7-sdk.zip Installationsdatei für die Sesame Datenbank

B.3 Online Quellen (PDF)

Pfad: /OnlineQuellen

DBpedia_About.pdf . . Beschreibung von DBpedia

Dbpedia_Spotlight_Webservice.pdf Beschreibung des DBpedia Spotlight Service

Dublin_Core_DCMI_Metadata_Terms.pdf DCMI Metadata Term-Spezifikation

FOAF_Vocabulary_Specification.pdf FOAF-Spezifikation

Linked_Data_DesignIssues.pdf Beschreibung zu Linked Data

LinkedOpenData.pdf . . Beschreibung zum Linked Open Data Projekt

Linking_Open_Data_Cloud.pdf Abbildung von der Linked Open Data Cloud

LOD-TheEssentials.pdf Linked Open Data: The Essentials

OpenDirectoryProject_About.pdf Beschreibung zum Open Directory Projekt

OWL_Guide.pdf . . . . OWL – Web Ontology Language Spezifikation

RDF_Primer.pdf . . . . RDF-Spezifikation

RDF_Semantic_Web_Standards.pdf Beschreibung zum RDF-Standard

SKOS_Introduction.pdf Beschreibung zu SKOS

SKOS_System_Reference.pdf SKOS-Spezifikation

WordNet_About_WordNet.pdf Beschreibung zu WordNet

Zemanta_Api_Companion.pdf Dokumentation von der Zemanta API

Zemanta_Dokumentation.pdf Allgemeine Dokumentation von

Zemanta_SuggestDokumentation.pdf Dokumentation zu der Methode *zemanta.suggest* der Zemanta API

Quellenverzeichnis

Literatur

[1] Sören Auer u. a. „DBpedia: a nucleus for a web of open data“. In: *Proceedings of the 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference*. ISWC'07/ASWC'07. Busan, Korea: Springer-Verlag, Nov. 2007, S. 722–735.

[2] Andrea Bauer. *Analyse von Semantic Web-APIs und deren Methoden*. Hagenberg, Austria, 2011.

[3] Tim Berners-Lee, James Hendler, Ora Lassila u. a. „The Semantic Web“. In: *Scientific american* 284.5 (Mai 2001), S. 28–37.

[4] C. Bizer. „The Emerging Web of Linked Data“. In: *IEEE Intelligent Systems* 24.5 (2009), S. 87–92.

[5] Christian Bizer, Tom Heath und Tim Berners-Lee. „Linked Data - The Story So Far“. In: *International Journal on Semantic Web and Information Systems* 5.3 (2009), S. 1–22.

[6] Christian Bizer u. a. „DBpedia - A crystallization point for the Web of Data“. In: *Web Semantics* 7.3 (Sep. 2009), S. 154–165.

[7] Uldis Bojars u. a. „Social Network and Data Portability using Semantic Web Technologies“. In: *2nd Workshop on Social Aspects of the Web (SAW 2008) at BIS2008*. 2008, 5–19.

[8] Dan Brickley und Libby Miller. *The Friend of a Friend (FOAF) Vocabulary Specification*. Aug. 2010.

[9] Peter Brusilovsky und Eva Millán. „User models for adaptive hypermedia and adaptive educational systems“. In: *The Adaptive Web*. Hrsg. von Peter Brusilovsky, Alfred Kobsa und Wolfgang Nejdl. Berlin, Heidelberg: Springer-Verlag, 2007, 3–53.

[10] Philip K. Chan. „Constructing Web User Profiles: A non-invasive Learning Approach“. In: *Revised Papers from the International Workshop on Web Usage Analysis and User Profiling*. WEBKDD '99. London, UK, UK: Springer-Verlag, 2000, S. 39–55.

[11] Paul Alexandru Chirita u. a. „Using ODP metadata to personalize search“. In: *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '05. New York, NY, USA: ACM, 2005, 178–185.

[12] Fefie Dotsika. „Semantic APIs: Scaling up towards the Semantic Web“. In: *International Journal of Information Management: The Journal for Information Professionals* 30.4 (Aug. 2010), S. 335–342.

[13] Susan Dumais u. a. „Stuff I've seen: a system for personal information retrieval and re-use“. In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*. SIGIR '03. Toronto, Canada: ACM, 2003, S. 72–79.

[14] Davide Eynard. *Using semantics and user participation to customize personalization*. Techn. Ber. HP Labs, 2008.

[15] Evgeniy Gabrilovich und Shaul Markovitch. „Overcoming the brittleness bottleneck using wikipedia: enhancing text categorization with encyclopedic knowledge“. In: *Proceedings of the 21st National Conference on Artificial Intelligence*. Bd. 2. AAAI'06. Boston, Massachusetts: AAAI Press, 2006, S. 1301–1306.

[16] Susan Gauch, Jason Chaffee und Alexander Pretschner. „Ontology-based personalized search and browsing“. In: *Web Intelligence and Agent Systems* 1 (2003), 1–3.

[17] Susan Gauch u. a. „User profiles for personalized information access“. In: *The Adaptive Web*. Hrsg. von Peter Brusilovsky, Alfred Kobsa und Wolfgang Nejdl. Berlin, Heidelberg: Springer-Verlag, 2007, 54–89.

[18] M Grcar, D Mladenic und M Grobelnik. „User Profiling for Interest-focused Browsing History“. In: ESWC'05/Heraklion'05. 2005.

[19] Pascal Hitzler u. a. *Semantic Web: Grundlagen (eXamen.press)*. 1. Aufl. Published: Paperback. Berlin, Heidelberg: Springer-Verlag, Okt. 2007.

[20] Hyoung R. Kim und Philip K. Chan. „Learning implicit user interest hierarchy for context in personalization“. In: *Proceedings of the 8th International Conference on Intelligent User Interface*. IUI '03. New York, NY, USA: ACM, 2003, 101–108.

[21] Alfred Kobsa. „Adaptive Verfahren – Benutzermodellierung“. In: *Grundlagen der Information und Dokumentation*. Hrsg. von R. Kuhlen, T. Seeger und D. Strauch. 5. Aufl. Munich: K. G. Saur, 2004.

[22] Chunliang Lu, Wai Lam und Yingxiao Zhang. „Twitter User Modeling and Tweets Recommendation Based on Wikipedia Concept Graph“. In: *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*. ITWP'12.

[23] Nicolaas Matthijs und Filip Radlinski. „Personalizing web search using long term browsing history“. In: *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*. WSDM '11. New York, NY, USA: ACM, 2011, 25–34.

[24] Bamshad Mobasher. „Data mining for web personalization“. In: *The Adaptive Web*. Hrsg. von Peter Brusilovsky, Alfred Kobsa und Wolfgang Nejdl. Berlin, Heidelberg: Springer-Verlag, 2007, 90–135.

[25] Michael Pazzani, Jack Muramatsu und Daniel Billsus. „Syskill & Webert: Identifying interesting web sites“. In: *Proceedings of the Thirteenth National Conference on Artificial Intelligence*. Bd. 1. AAAI'96. Portland, Oregon: AAAI Press, 1996, 54–61.

[26] Pellegrini und Blumauer. *Semantic Web Wege Zur Vernetzten Wissensgesellschaft ; Mit 4 Tabellen*. X.media.press. Berlin, Heidelberg: Springer-Verlag, 2006.

[27] Dimitrios Pierrakos u. a. „Web Usage Mining as a Tool for Personalization: A Survey“. In: *User Modeling and User-Adapted Interaction* 13.4 (Nov. 2003), 311–372.

[28] Krishnan Ramanathan und Komal Kapoor. „Creating User Profiles Using Wikipedia“. In: *Proceedings of the 28th International Conference on Conceptual Modeling*. ER '09. Berlin, Heidelberg: Springer-Verlag, 2009, 415–427.

[29] Gerard Salton und Michael J. McGill. *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc., 1986.

[30] Kazunari Sugiyama, Kenji Hatano und Masatoshi Yoshikawa. „Adaptive web search based on user profile constructed without any effort from users“. In: *Proceedings of the 13th International Conference on World Wide Web*. WWW '04. New York, NY, USA: ACM, 2004, 675–684.

[31] Pu Wang u. a. „Using Wikipedia knowledge to improve text classification“. In: *Knowledge and Information Systems* 19.3 (Mai 2009), S. 265–281.

[32] Yabo Xu u. a. „Privacy-enhancing personalized web search“. In: *Proceedings of the 16th International Conference on World Wide Web*. WWW '07. New York, NY, USA: ACM, 2007, 591–600.

[33] Huirong Yang u. a. „A Semantic Similarity Measure between Web Services Based on Google Distance“. In: *Proceedings of the 2011 IEEE 35th Annual Computer Software and Applications Conference*. COMPSAC '11. Washington, DC, USA: IEEE Computer Society, 2011, 14–19.

Online-Quellen

[34] *DBpedia – About*. URL: http://dbpedia.org/About (besucht am 26.09. 2013).

[35] *DBpedia Spotlight Webservice*. URL: https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki/Web-service (besucht am 26.09.2013).

[36] *Dublin Core - DCMI Metadata Terms*. URL: http://dublincore.org/documents/dcmi-terms/ (besucht am 26.09.2013).

[37] *Linked Data*. URL: http://www.w3.org/DesignIssues/LinkedData.html (besucht am 26.09.2013).

[38] *Linked Open Data – The Essentials*. URL: LOD-TheEssentials.pdf (besucht am 26.09.2013).

[39] *Linking Open Data cloud diagram*. URL: http://lod-cloud.net/ (besucht am 26.09.2013).

[40] *ODP - Open Directory Project*. URL: http://www.dmoz.org/docs/en/about.html (besucht am 23.05.2013).

[41] *OWL Web Ontology Language Guide*. URL: http://www.w3.org/TR/owl-guide/ (besucht am 26.09.2013).

[42] *RDF - Semantic Web Standards*. URL: http://www.w3.org/RDF/ (besucht am 24.04.2013).

[43] *Resource Description Framework (RDF). W3C Recommendation*. URL: http://www.w3.org/TR/rdf-primer/.

[44] *SKOS Simple Knowledge Organization System Reference*. URL: http://www.w3.org/TR/2009/REC-skos-reference-20090818/ (besucht am 26.09.2013).

[45] *SPARQL Protocol for RDF. W3C Recommendation*. URL: http://www.w3.org/TR/rdf-sparql-protocol/ (besucht am 13.05.2013).

[46] *Simple Knowledge Organization System - Introduction to SKOS*. URL: http://www.w3.org/2004/02/skos/intro (besucht am 26.09.2013).

[47] *The Friend of a Friend (FOAF) Vocabulary Specification*. URL: http://xmlns.com/foaf/spec/ (besucht am 10.08.2013).

[48] *WordNet - WordNet - About WordNet*. URL: http://wordnet.princeton.edu/ (besucht am 18.06.2013).

[49] *Zemanta - Documentation of method* zemanta.suggest. URL: http://developer.zemanta.com/docs/suggest/ (besucht am 26.09.2013).

[50] *Zemanta*. URL: http://developer.zemanta.com/docs/ (besucht am 26. 09.2013).

[51] *Zemanta*. URL: http://developer.zemanta.com/media/files/docs/zemanta_api_companion.pdf (besucht am 26.09.2013).