Universität Regensburg

Aufbau eines wissenschaftlichen Textcorpus auf der Basis der Daten der englischsprachigen Wikipedia

Masterarbeit im Fach Informationswissenschaft
Institut für Information und Medien, Sprache und Kultur

von:	Markus Fuchs
Adresse:	Lährer Weg 97 92318 Neumarkt
Matrikelnummer:	13 823 49
Erstgutachter:	Prof. Dr. Christian Wolff
Zweitgutachter:	Prof. Dr. Rainer Hammwöhner
Laufendes Semester:	Winter-Semester 2009/10
Abgabedatum:	15. Dezember 2009

Zusammenfassung

Die Wikipedia hat sich in den letzten Jahren zu einer vielversprechenden Forschungs-Ressource entwickelt. Ihr enzyklopädischer Aufbau, ihre freie Verfügbarkeit und die Aktualität der Inhalte sind nur ein Teil der Gründe, die die Online-Enzyklopädie so attraktiv für viele wissenschaftliche Bereiche (*Information Retrieval*, *Information Extraction*, natürliche Sprachverarbeitung, maschinelles Lernen, ...) machen.

Doch der Zugriff auf die in der Wikipedia enthaltenen Informationen ist nicht leicht, da sie in Wikitext, der Wikipedia-eigenen Markup-Sprache, enkodiert sind. Die maschinelle Verarbeitung von Wikitext-Markup ist jedoch sehr schwer, weil eine formale Definition fehlt.

Diese Arbeit beschreibt ein System, das aus den Daten der englischen Wikipedia automatisch ein Textcorpus erstellen kann, das die häufigsten zu Forschungszwecken verwendeten Inhalte enthält. Bei der Erstellung des Corpus werden die Artikeltexte darüber hinaus mit Lemma- und Part-of-Speech-Informationen annotiert und Kookkurrenz-Häufigkeiten extrahiert. Wegen der Speicherung aller Daten in einer relationalen Datenbank ist ein sehr effizienter Zugriff auf die Wikipedia-Daten mit umfassender Suchfunktionalität möglich.

Abstract

With the growth in popularity over the last eight years, Wikipedia has become a very promising resource in academic studies. Some of its properties make it attractive for a wide range of research fields (information retrieval, information extration, natural language processing, ...), e. g. free availability and up to date content.

However, efficient and structured access to this information is not easy, as most of Wikipedia's contents are encoded in its own markup language. And, unfortunately, there is no formal definition of wikitext, which makes parsing very difficult and burdensome.

In this thesis, we present a system that lets the researcher automatically build a richly annotated corpus containing the information most commonly used in research projects. To this end, we built our own wikitext parser based on the original converter used by Wikipedia itself to convert wikitext into HTML. The system stores all data in a relational database, which allows for efficient access and extensive retrieval functionality.

Inhaltsverzeichnis

1. Einleitung

Im Herbst 1999 hatte Jimmy Wales die Idee zur Gründung einer freien, kollaborativ erstellten Online-Enzyklopädie. Die Artikel sollten von Experten auf freiwilliger Basis geschrieben und ihre Qualität durch ein Peer-Review-Verfahren sichergestellt werden. Anfang des Jahres 2000 ging das „Nupedia“ genannte Projekt unter der Führung von Larry Sangers an den Start und gegen Mitte des Jahres wurde der erste Artikel veröffentlicht[1] (vgl. Slashdot, 2005).

Jimmy Wales und Larry Sangers erkannten sehr schnell, dass das strikte Review-Verfahren nicht gerade der Produktivität förderlich war. Deshalb schlugen sie im Januar 2001 vor, ein Wiki zu veröffentlichen, „where [anybody] can simply write down ideas, post articles, etc.“ (Sangers, 2001), dessen Inhalte dann Stück für Stück in die Nupedia aufgenommen werden sollten.

Heute, mehr als acht Jahre später, ist aus diesem „Nebenprodukt“ eine der zehn meist besuchten Internetseiten der Welt geworden. Täglich rufen durchschnittlich mehr als 300 Millionen Nutzer die Seiten der Wikipedia auf[2].

Parallel zu dieser Entwicklung wuchs auch das wissenschaftliche Interesse an der Wikipedia seit 2001 stetig an. Forscher aus verschiedenen Fachbereichen nutzen die reiche Fülle an frei verfügbaren Informationen für ihre Untersuchungen. Allerdings ist die direkte und effiziente Verarbeitung der Daten sehr schwierig, weil die meisten Informationen nur in semi-strukturierter Form im eigenen Markup-Format, genannt „Wikitext“, vorliegen und sehr umfangreich sind. Allein die Inhalte aller aktuellen Artikel der englischen Wikipedia benötigen bereits mehr als 20 GB Speicherplatz. Nimmt man die komplette Versionsgeschichte aller Artikel hinzu, wächst die Datenmenge auf mehrere Terabytes an. De Alfaro und Ortega sprechen in diesem Zusammenhang deshalb vom „Wikipedia data jungle“ (de Alfaro und Ortega, 2009, S. 1).

Wir wollen in dieser Arbeit die Daten in eine strukturiertere Form bringen. Das Hauptaugenmerk liegt dabei auf Anwendungen im Bereich der natürlichen Sprachverarbeitung, indem wir vor allem den Zugriff auf die Artikeltexte erleichtern. Um aber möglichst viele Forschungsvorhaben zu ermöglichen, sollen auch eher strukturelle Daten wie die Artikel-Verlinkungen in die Datensammlung aufgenommen werden.

Ziel dieses Projektes ist deshalb die Entwicklung eines Systems, mit dessen Hilfe man automatisiert ein auf den Daten der englischen Wikipedia basierendes Corpus erstellen kann, um so dazu beizutragen, den „Daten-Dschungel zu lichten“.

Der nächste Abschnitt untersucht dazu einige wissenschaftliche Arbeiten mit der Wiki-

[1] „Atonality“ von Christoph Hust
[2] http://wikistics.falsikon.de/latest/

pedia. Dabei interessiert uns vor allem die Frage, welche ihrer Eigenschaften sie so reizvoll für die Wissenschaft machen. Außerdem wollen wir herausfinden, welche in der Wikipedia enthaltenen Daten für die verschiedenen akademischen Studien benötigt werden. In Abschnitt 3 werden dann einige vergleichbare Arbeiten vorgestellt. Das nachfolgende Kapitel beschreibt den Entwurf und die Implementierung unseres Systems genauer. Dieses System wir dann in Abschnitt 5 einer Bewertung unterziehen und zeigen, wie wissenschaftliche Untersuchungen von unserem Corpus profitieren können. Wir schließen die Arbeit mit einem Fazit in Abschnitt 6.

2. Wissenschaftliche Forschung

Wie bereits erwähnt, ist das wissenschaftliche Interesse an der Wikipedia in den letzten Jahren stark gewachsen. (Abbildung 1 zeigt die Entwicklung von 2001 bis 2007.) Ein großer Teil der Untersuchungen verwendet dabei die Daten der Wikipedia, um andere Forschungsprobleme zu lösen. Zudem bildet sich zunehmend ein eigener Forschungsstrang heraus, der direkt die Wikipedia selbst untersucht, die sogenannte „Wikipedistik“ (vgl. König, 2009, S. 33).

Um zu verstehen, was die Wikipedia für die Wissenschaft so interessant macht, stellt der nächste Abschnitt einige Merkmale der Online-Enzyklopädie genauer vor.

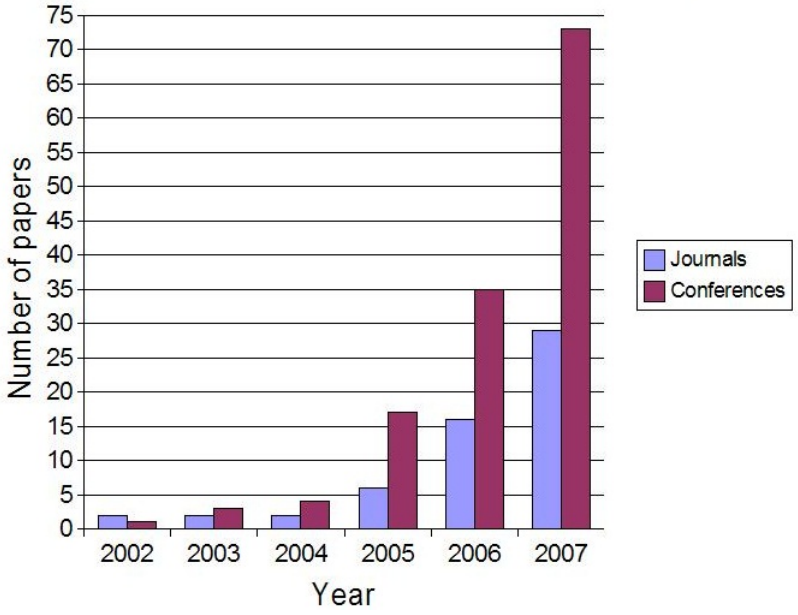

Abbildung 1: Entwicklung des akademischen Interesses an der Wikipedia (Wikipedia, 2008c)

2.1. Charakteristika der Wikipedia

Wohl eine der wichtigsten Eigenschaften der Wikipedia, die zu dem akademischen Interesse beitragen, ist die Tatsache, dass alle Inhalte *frei verfügbar* sind. Dadurch lassen sich die Daten beispielsweise relativ leicht als Teil einer Testsammlung verteilen (vgl. Sigurbjörnsson u. a., 2006, S. 1). Als Lizenz wird die GFDL (*GNU Free Documentation License*) verwendet, eine Sonderform der GPL (*GNU Public License*), die eigentlich für die Dokumentation von OS-Software gedacht ist. Ein Vorteil dieser Lizenz ist, dass auch eine kommerzielle Verwendung der Daten nicht ausgeschlossen ist.

Eine weitere Eigenschaft der Wikipedia ist ihre *große Nutzergemeinde*, die mit mehr als 5 Millionen registrierten Benutzern vermutlich eine der größten der Welt ist. Das macht sie vor allem für soziokulturelle Studien interessant, in denen beispielsweise versucht wird, bestimmte Verhaltensmuster zu identifizieren (vgl. Ortega, 2009, S. 4). Zudem ist es die große Menge an freiwilligen Mitarbeitern, die die weite *Abdeckung* von Wissensgebieten möglich macht. Zusammen mit der damit einhergehenden *Aktualität* der Daten sind dies zwei weitere Merkmale, die vor allem auf dem Gebiet der Wissensextraktion von Vorteil sind (vgl. Syed u. a., 2008, S. 137).

Vorteilhaft z. B. für die Verwendung beim „Information Retrieval“ ist die Tatsache, dass es sich bei der Wikipedia um eine Enzyklopädie handelt und sie deshalb auch dementsprechend aufgebaut ist (*Enzyklopädität)*: Jeder Artikel beschreibt einen Begriff; Weiterleitungsseiten bilden verschiedene Schreibweisen auf den gleichen Begriff ab; auf Disambiguierungsseiten wird zwischen mehreren Begriffen unterschieden, die durch den gleichen Term beschrieben werden. Die *dichte Verweisstruktur*, sowohl zwischen den Artikeln als auch durch die Kategorienhierarchie, macht aus der Enzyklopädie ein (semantisches) Text-Netzwerk (vgl. Mehler, 2008, S. 329).

Ein letztes Merkmal ist durch die Verfügbarkeit in mehr als 250 Sprachen gegeben (*Multilingualität*). Dass es möglich ist, auf inhaltsgleiche Artikel in anderen Sprachen zu verweisen, macht die Wikipedia interessant sowohl für multilinguales Retrieval als auch für Anwendungen im Bereich des maschinellen Übersetzens (vgl. Adafre und de Rijke, 2006).

2.2. Verwendete Daten

Um herauszufinden, welche Daten der Wikipedia zu Forschungszwecken genutzt werden, haben wir mehr als 40 zufällig ausgewählte Veröffentlichungen aus den verschiedensten Bereichen untersucht (*Information Retrieval*, *Information Extraction*, *Wikipedistik*, *Machine Translation*, ...). Die Ergebnisse dieser Literaturanalyse werden wir im Nach-

folgenden vorstellen und dabei auf einige Arbeiten näher eingehen.[3]

2.2.1. Artikeltext

Die wohl offensichtlichsten Informationen in der Wikipedia, die zur Forschung genutzt werden können, sind die Artikeltexte. Sie wurden in mehr als 20 Studien verwertet und sind damit der am häufigsten verwendete Wikipedia-Inhalt unserer Untersuchung.

Wir haben oben bereits angedeutet, dass die Enzyklopädität – ein Artikel beschreibt einen semantischen Begriff (*Signifikat*) – von Vorteil ist. Dadurch lassen sich die Artikeltexte als Begriffsdefinitionen interpretieren und deren Titel als das auf den Begriff verweisende Zeichen (*Signifikant*). Die Arbeiten von Ruiz-Casado u. a. (2005a,b) gehören zu den ersten Untersuchungen, die sich diese Tatsache zu Nutze machen. Die Autoren stellen ein Verfahren vor, mit dem Wikipedia-Einträge automatisiert auf die Begriffe von WordNet abgebildet werden können. Dazu erstellen sie Term-Vektoren sowohl für die Artikeltexte als auch für die Glossen potenziell bedeutungsgleicher WordNet-Synsets und berechnen deren Ähnlichkeit. Basierend auf diesem System, identifizieren sie in (Ruiz-Casado u. a., 2005b) Phrasen, die die semantischen Beziehungen Hypernymie, Hyponymie, Meronymie und Holonymie in natürlicher Sprache ausdrücken (z. B. „X is part of Y“). Die gefundenen Textmuster verwenden sie dann, um ein bestehendes semantisches Netzwerk zu erweitern.

Auch das von Gabrilovich und Markovitch vorgestellte *ESA (Explicit Semantic Analysis)*-Verfahren nützt den enzyklopädischen Aufbau der Wikipedia aus. Sie definieren einen Artikel als Begriffsdefinition: „[W]e view each Wikipedia article as defining a concept that corresponds to each topic“ (Gabrilovich und Markovitch, 2009, S. 447). Dann berechnen sie für jeden Artikel den TF-IDF (*term frequency – inverse document frequency*)-Wert und erhalten so für jeden in der Wikipedia enthaltenen Begriff eine Repräsentation in Form eines Term-Vektors. Diese Vektoren verwenden sie dann u. a. um die semantische Verwandtheit von Texten zu bestimmen.

Ramanathan u. a. (2009) nutzen die Wikipedia, um Textzusammenfassungen zu erstellen. Dazu indizieren sie zuerst alle Artikel mit einer Volltextsuchmaschine (*Lucene*). Dieser übergeben sie dann die einzelnen Sätze des Eingabedokuments als Abfrage, ermitteln den Artikeltitel der Ergebnisse und bilden so jeden Satz des Dokuments auf einen Wikipedia-Begriff ab. Daraufhin bestimmen sie entweder den häufigsten Begriff oder alle über einem festgelegten Grenzwert und wählen diejenigen Sätze für die Zusam-

[3] Es soll uns an dieser Stelle die Präsentation einer kleinen Auswahl genügen, zumal viele der Studien die gleichen Daten verwendet haben. Im Anhang B findet sich eine Tabelle mit allen Ergebnissen der Literaturanalyse.

menfassung aus, die mit diesen Begriffen verknüpft sind. Die Autoren haben ihr System mit den Testdaten der *Document Understanding Conference (DUC) 2002* evaluiert und hätten dort den dritten Platz erreicht (vgl. Ramanathan u. a., 2009, S. 259).

2.2.2. Kategorien-Hierarchie

Seit Mai 2004 erlaubt es die Wikipedia, mehrere Artikel in einer Kategorie zusammenfassen. Diese lassen sich wiederum selbst in Kategorien einteilen. Dabei bildet sich allerdings kein Baum, sondern ein gerichteter Graph, weil eine Kategorie mehrere Ober-Kategorien haben kann (vgl. Schönhofen, 2006, S. 456). Die Artikel entsprechen dann den Endknoten des Graphen.

Voss hat in seinen informetrischen Untersuchungen der Wikipedia festgestellt, dass das Kategoriensystem „eine spezielle Form des ‚social tagging' [ist], die als Besonderheit Elemente einer Klassifikation beinhaltet", da „die Vergabe von Kategorien ohne einheitliches Regelwerk stattfindet" (Voss, 2005a, S. 22).

Das hat natürlich Auswirkungen auf die Qualität des Ordnungssystems. Studien von Hammwöhner haben gezeigt, dass viele Kategorien entweder zu allgemein oder zu spezifisch sind und wichtige Kategorien fehlen (vgl. Hammwöhner, 2007b, S. 18). Erschwerend kommt hinzu, dass die Kategorien-Hierarchie auch zur Administration verwendet wird. So ist die Kategorie „articles lacking sources" mit 146.932 Artikeln die zweitgrößte in der englischen Wikipedia.

Trotz seiner Schwächen findet das Kategoriensystem immer wieder Anwendung in akademischen Studien. Prominentestes Beispiel sind die Arbeiten von Ponzetto und Strube (2007). Sie verwenden eine syntaktische Analyse der Titel verknüpfter Kategorien, um deren semantische Beziehung zu bestimmen. Dabei beschränken sie sich auf die *isa*- und *notisa*-Relationen. Zusätzlich beziehen sie das gesamte Corpus aller Wikipedia-Artikeltexte mit ein, um weitere Relationen – ähnlich dem Ansatz von Ruiz-Casado u. a. (2005b) – über einen Textmuster-Abgleich zu finden. Die so erstellte Taxonomie verwenden sie dann, um die semantische Ähnlichkeit von zwei Wörtern zu berechnen. In Ponzetto und Navigli (2009) stellen sie eine Methode vor, um die Taxonomie in WordNet zu integrieren.

Zirn u. a. (2008) erweitern das von Ponzetto und Strube (2007) beschriebene Verfahren um die Möglichkeit, Instanzen von Klassen zu unterscheiden. Und Kassner u. a. (2008) übertragen es auf die deutsche Wikipedia-Ausgabe und vergleichen die erhaltene Taxonomie mit GermaNet, der deutschen Version von WordNet.

2.2.3. Zwischensprachliche Verweise

Wie bereits erwähnt, gibt es die Wikipedia in mehreren hundert Sprachversionen, die untereinander verlinkt sind. Aus den zwischensprachlichen Verweisen lässt sich zwar kein paralleles Corpus erstellen, da die Artikel in den anderen Sprachen keine Übersetzungen sind, sondern nur den gleichen Begriff beschreiben[4]. Doch auch diese sogenannten „vergleichbaren Corpora“ lassen sich vielseitig einsetzen.

De Smet und Moens (2009) beschreiben eine Methode, mit der sich bestimmen lässt, ob zwei Nachrichtenmeldungen in verschiedenen Sprachen über das gleiche Ereignis berichten. Dazu haben sie 7612 Artikel aus der niederländischen Wikipedia und deren englische Entsprechungen zufällig ausgewählt. Sie setzen dieses Corpus erfolgreich ein, um damit ein *Latent Dirichlet Allocation (LDA)*-Modell zu trainieren, das es ermöglicht ein Dokument als Wahrscheinlichkeitsverteilung von Themen zu beschreiben. Sie erweitern den Algorithmus dahingehend, dass zwei Themen-Mengen in unterschiedlichen Sprachen gleichzeitig gelernt werden können, damit sich die Themenverteilungen für beide Sprachen miteinander vergleichen lassen.

Auch Potthast u. a. (2008) nutzen die Wikipedia als „vergleichbares Corpus“. Sie verwenden es, um zu einem gegebenen Text in einer Sprache Dokumente einer anderen Sprache mit gleichem Inhalt zu finden. Sie schlagen ihre Methode zur Plagiats-Erkennung vor.

2.2.4. Infoboxen

Einhergehend mit der Idee des *Semantic Web* wird auch ein *Semantic Wiki* und dessen Anwendung auf die Wikipedia vorgeschlagen, um die darin enthaltenen Informationen auch für Computer verstehbar zu machen. Dabei wird übersehen, dass die Darstellung von strukturierten Inhalten auch jetzt bereits möglich ist und genutzt wird. Sogenannte Infoboxen, eine bestimmte Art von Vorlagen, ermöglichen es, Daten über Attribut-Wert-Paare einzugeben, die dann automatisch auf der Seite gerendert werden.

Das wohl bekannteste System, das davon Gebrauch macht, ist *DBpedia*[5]. Die Ersteller dieses Systems extrahieren aus allen Artikelseiten der Wikipedia die Attribut-Wert-Paare aller Infoboxen und konvertieren die darin enthaltene Information ins RDF-Format (Auer u. a., 2007).

Eine spezifischere Anwendung von Infoboxen stellen Athenikos und Lin (2009) vor.

[4] Zumindest ist das im Allgemeinen der Fall. Untersuchungen von Hammwöhner haben ergeben, dass zum einen Fehlverweise vorkommen und zum anderen bis zu 5 % der Verweise inkonsistent sind (vgl. Hammwöhner, 2007a, S. 6) (d. h. es existiert nur ein Link in eine Richtung).

[5] *http://dbpedia.org/*

Sie untersuchen die Wikipedia-Einträge von 300 wichtigen Philosophen aus dem „Timeline of Western Philosophers"-Artikel der englischen Wikipedia. Daraus extrahieren sie unter anderem die in den Infoboxen abgebildeten Werte für „influenced" und „influenced by". Die so gefundenen Beziehungen von Philosophen untereinander und philosophischen Begriffen visualisieren sie auf ihrer Homepage[6].

2.2.5. Hyperlinks

In einem Wiki ist es sehr leicht, auf andere darin enthaltene Seiten zu verweisen. Normalerweise reicht es, den Titel des Link-Ziels in doppelte eckige Klammern zu setzen. Dementsprechend sind auch die Artikel in der Wikipedia stark untereinander verlinkt. Generell ist davon auszugehen, dass die Begriffe von zwei miteinander verknüpften Artikeln irgendeine (semantische) Beziehung zueinander haben. Gabrilovich und Markovitch (2009) weisen jedoch darauf hin, dass ihre Untersuchungen ergeben haben, dass es oft vorkommt, dass zwei Begriffe nicht wirklich verwandt sind, obwohl ihre Artikel miteinander verlinkt sind. So hat beispielsweise der „Education"-Abschnitt des „United States"-Artikels viele Verweise auf die „High School"-, „College"- und „Literacy Rate"-Artikel (vgl. Gabrilovich und Markovitch, 2009, S. 449).

Die Hyperlink-Struktur bildet, genau wie die Kategorien-Hierarchie, einen gerichteten Graphen, bei der die Links die Kanten und die Artikel die Knoten sind. Eine ganze Reihe von Studien analysiert dessen Aufbau. Zlatic u. a. (2006) gehen davon aus, dass sich die Link-Graphen unterschiedlicher Sprachversionen in verschiedenen Wachstumsstufen befinden. Deshalb vergleichen sie die Graphen mehrerer Sprachen, um so Wissen über Wachstumsprozesse von komplexen Netzwerken zu gewinnen. Capocci u. a. (2006) stellen bei ihren Untersuchungen fest, dass das Wachstum des Wikipedia-Graphen – wie der des WWW – den Regeln des *Preferential Attachment („richer-get-richer"*-Regel*)* folgt, d. h. Artikel, die bereits viele eingehende Links besitzen, erhalten mit der Zeit noch mehr Verweise. Buriol u. a. (2006) verwenden die in Abschnitt 2.2.7 beschriebene Versionsgeschichte aller Artikel, um das Wachstum des Graphen im Zeitverlauf zu untersuchen.

Ito u. a. (2008) nutzen die in den Artikeln vorkommenden Link-Kookkurrenzen, um die semantische Verwandtheit von zwei Begriffen zu bestimmen. Sie sehen die Vorteile ihrer Methode in der Skalierbarkeit: „[it is] more scalable than link structure analysis because it is a one-pass process" (Ito u. a., 2008, S. 817).

Die in einem Artikel vorkommenden Links finden vor allem bei Arbeiten zur *Named Entity Detection* und *Word Sense Disambiguation* noch eine ganz andere Verwen-

[6] http://research.cis.drexel.edu:8080/sofia/WPS/

dung. Die Markup-Sprache der Wikipedia erlaubt auch die Auszeichnung von Link-Bezeichnern. Dieser sogenannte „Ankertext“ erscheint dann an Stelle des Titels des verlinkten Artikels im Text. In Verbindung damit lässt sich der Link als Begriffsannotation interpretieren. Mihalcea (2007) macht sich genau das zu Nutze und erstellt aus der Wikipedia ein bedeutungsannotiertes Corpus. Dieses verwendet er dann zum Trainieren eines naiven Bayes-Klassifikators zur *Word Sense Disambiguation*.

2.2.6. Weiterleitungs- und Disambiguierungsseiten

Eine spezielle Form von Artikeln sind die Disambiguierungsseiten. Auf ihnen werden die verschiedenen Bedeutungen polysemer Ausdrücke – zumeist in Listen-Form – zusammengefasst. Von dort wird dann über einen Link auf den jeweiligen Artikel verwiesen.

Die Wikipedia erlaubt es, über einen Weiterleitungsmechanismus mehrere Artikeltitel auf einen Artikel abzubilden. Dadurch ist es möglich, von anderen Schreibweisen oder Abkürzungen auf den Hauptartikel zu verweisen. Weiterleitungsseiten werden in den wissenschaftlichen Untersuchungen hauptsächlich dazu verwendet, um Synonyme zu finden.

So nutzt beispielsweise Kinzler (2008) unter anderem sowohl Weiterleitungs- als auch Disambiguierungsseiten bei der Erstellung eines multilingualen Thesaurus.

2.2.7. Versionsgeschichte

Die Tatsache, dass bei einem Wiki alle Versionen eines Artikels gespeichert werden und weiterhin darauf zugegriffen werden kann, ist vor allem für Forschungsprojekte interessant, die versuchen, die Qualität eines Artikels automatisch zu bestimmen. Wenn man davon ausgeht, dass ein Artikel mit der Zeit immer besser wird, lassen sich aus seiner Versionsgeschichte vielleicht interessante Schlüsse ziehen.

So untersuchen Wöhner und Peters (2009) den gesamten Lebenszyklus eines Artikels der deutschen Wikipedia von seiner Entstehung bis hin zur Markierung als Löschkandidat oder als guter (*good*) oder sehr guter (*featured*) Artikel. Sie unterteilen die Beiträge zu einem Artikel in zwei Gruppen: vorübergehend (*transient*) und dauerhaft (*persistent*). Als „vorübergehend“ klassifizieren sie Beiträge, die binnen kürzester Zeit (weniger als drei Minuten) wieder „revertiert“ werden. „Dauerhafte“ Änderungen bei einem Artikel bleiben über einen langen Zeitraum hinweg im Artikel enthalten und gelten als akzeptiert von der Wikipedia-Gemeinde. Die Vermutung der Autoren, dass sich die Werte für vorübergehende und dauerhafte Beiträge bei guten und schlechten Artikeln stark unterscheiden, wird in ihren Untersuchungen bestätigt. Sie schlagen deshalb einige Metriken

zur automatischen Bestimmung der Artikelqualität vor, die auf dieser Beobachtung beruhen.

Auch Hu u. a. (2007) wollen die Qualtität eines Artikels automatisch bestimmen. Dazu ermitteln sie für jedes Wort einer Artikelversion, von welchem Autor es stammt. Die von ihnen vorgeschlagenen Metriken beruhen auf den Annahmen, dass gute Autoren viele Wörter zu qualitativ hochwertigen Artikeln beigetragen haben und umgekehrt, dass bei einem guten Artikel viele Beiträge von guten Autoren stammen. Zusätzlich beziehen sie mit ein, wie viele Wörter eines Artikels von guten Autoren überprüft wurden. Als „überprüft“ gelten alle Wörter eines Artikels, die von einer Artikelversion zur nächsten nicht verändert wurden. Ihr Verfahren zur Bestimmung der Artikelqualität setzen sie dann ein, um Suchergebnisse danach zu sortieren.

Auch Ganjisaffar u. a. (2009) haben das Ziel, die Qualität eines Artikels in das Ranking einfließen zu lassen. Dazu bestimmen sie für alle Artikel die Anzahl distinkter Autoren. Diese einfache Metrik konnte in ihren Studien die Qualität der Suchergebnis-Sortierung erfolgreich verbessern (vgl. Ganjisaffar u. a., 2009, S. 2).

Ortega (2009) verwendet die Versionshistorie von zehn Wikipedia-Versionen[7] für umfassende quantitative Studien. Darin untersucht er unter anderem die soziale Struktur und die Reputation von Autoren und analysiert die Demographie der Wikipedia-Gemeinde (vgl. Ortega, 2009, S. 52).

Unsere Literaturanalyse hat gezeigt, dass viele der in der Wikipedia und deren Aufbau implizit enthaltenen Informationen zur Forschung verwendet werden. Der nächste Abschnitt wird nun einige ähnliche Projekte wie das hier beschriebene vorstellen, die ebenso einen einfacheren Zugriff auf diese Inhalte ermöglichen wollen. Dabei ist vor allem von Interesse, welche der zur Forschung verwendeten Daten, die wir in diesem Abschnitt ausfindig machen konnten, von den unterschiedlichen Systemen berücksichtigt werden.

3. Vergleichbare Arbeiten

Es wurde oben bereits erwähnt, *dass* alle Daten der Wikipedia frei verfügbar sind. Wir haben aber noch nicht angesprochen, *wie* man an diese Daten gelangen kann. Dazu bieten sich mehrere Möglichkeiten, die wir kurz vorstellen möchten.

Prinzipiell ist es natürlich immer möglich, die Daten auf dem gleichen Wege herunterzuladen wie der einfache Benutzer: über direkten Zugriff auf die Homepage. Allerdings würde der automatisierte Abruf aller Inhalte eine enorme Last auf die Server ausüben,

[7] Englisch, Deutsch, Französisch, Polnisch, Japanisch, Niederländisch, Italienisch, Portugiesisch, Schwedisch und Spanisch (vgl. Ortega u. a., 2008, S. 305)

weshalb davon abgeraten wird[8]. Werden die Daten dennoch im HTML-Format benötigt, können sie über sogenannte „Static Dumps“ bezogen werden[9], die allerdings nicht sehr oft aktualisiert werden[10].

Eine weitere Möglichkeit, die Wikipedia-Daten zu erhalten, ist der Zugriff über die „Datenbank-Backup-Dumps“. Sie werden in mehr oder weniger regelmäßigen Abständen (alle 1–3 Wochen) für alle Sprachversionen erstellt und unter http://download.wikimedia.org zum Herunterladen zur Verfügung gestellt. Die Artikeltexte befinden sich im Wikitext-Format in XML-Dateien. Diese gibt es in verschiedenen Versionen, die sich in der Menge der jeweils darin enthaltenen Daten unterscheiden:

- Alle Seiten mit der kompletten Versionshistorie (Größe: ca. 2 TB)
- Alle Seiten (u. a. auch Diskussions- und Benutzerseiten) in der aktuellen Version (Größe: ca. 50 GB)
- Nur die Artikel (auch Weiterleitungen), Kategorien, Templates (Größe: ca. 20 GB)

Zusätzlich finden sich dort noch einige der Datenbank-Tabellen im SQL-Format (z. B. die Liste aller Artikelnamen oder geschützten Artikel).

Nun sind aber weder HTML noch Wikitext gut zur maschinellen Verarbeitung geeignet[11]. Um trotzdem einen leichten Zugriff auf die große Fülle an Informationen zu gewähren, gibt es bereits einige Systeme bzw. Ressourcen, die wir im Folgenden kurz vorstellen möchten.

3.1. *Wikipedia XML Corpus*

Das *Wikipedia XML Corpus* von Denoyer und Gallinari enthält die Artikel von acht verschiedenen Sprachversionen der Wikipedia (Englisch, Französisch, Deutsch, Niederländisch, Spanisch, Chinesisch, Arabisch, Japanisch). Jeder der Artikel ist dabei in einer eigenen XML-Datei gespeichert. Die Autoren haben dazu das Wikitext-Markup jedes Artikels in ein zu diesem Zweck erstelltes XML-Format umgewandelt, bei dem die verschiedenen Tags den Markup-Elementen von Wikitext entsprechen (z. B. `emph2/emph3` für Hervorhebungen, `collectionlink` für interne Verweise oder `title` für Überschriften). Die Information, in welchen Kategorien sich die Artikel befinden, ist in drei gesonderten Dateien gespeichert.

[8] http://en.wikipedia.org/wiki/Wikipedia:Database_download#Please_do_not_use_a_web_crawler
[9] http://static.wikipedia.org/
[10] Der zur Zeit aktuellste verfügbare ist vom Juni 2008.
[11] Warum dem so ist, werden wir in Abschnitt 4.5 näher erläutern

Sprache	Anzahl der Dokumente
Englisch	659.388
Französisch	110.838
Deutsch	305.099
Niederländisch	125.004
Spanisch	79.236
Chinesisch	56.661
Arabisch	11.637
Japanisch	187.492

Tabelle 1: Inhalt des *Wikipedia XML Corpus (vgl. Denoyer und Gallinari, 2006, S. 13)*

Verwendungsmöglichkeiten sehen die Autoren beispielsweise beim *XML Ad-hoc Retrieval*, zum Training und Test von Kategorisierungs-Algorithmen oder zum *Clustering* (vgl. Denoyer und Gallinari, 2006, S. 12). Das Corpus wird seit 2006 jedes Jahr bei der *INEX (INitiative for the Evaluation of XML Retrieval)* und bei der *XML Document Mining Challenge* eingesetzt.

Im Gegensatz zu den meisten anderen Arbeiten, die wir in den nächsten Abschnitten vorstellen werden, handelt es sich bei dem Corpus von Denoyer und Gallinari um eine Ressource. Das heißt, dass es nicht möglich ist, selbst ein Corpus basierend auf einem neueren Datenbank-Dump zu erstellen. Dadurch fällt einer der wichtigsten Vorteile der Wikipedia weg: ihre Aktualität. Wie man an Tabelle 1 sehen kann, ist das Corpus mittlerweile nicht mehr sehr aktuell, denn die englische Wikipedia enthält inzwischen mehr als vier Mal so viele Artikel.

Da die Markup-Elemente von Wikitext direkt auf XML-Elemente abgebildet wurden, enthält das Corpus alle wichtigen Daten. Der Zugriff darauf ist jedoch durch die Verwendung von XML als Speicherformat ohne weitere Vorkehrungen nicht sehr effizient. Allerdings wurde das Corpus ja eben gerade erstellt, um als Test- und Trainingsmenge von XML-Retrieval-Verfahren zu dienen.

3.2. *WikiPrep*

WikiPrep ist ein von Evgeniy Gabrilovich erstelltes Perl-Skript[12], das den Datenbank-Dump von Wikipedia in ein leichter zu verarbeitendes XML-Format verwandelt. Bei der Umwandlung wird der gesamte Wikitext-Markup bis auf die Überschriften aus dem Artikeltext entfernt und gesondert gespeichert. Die Überschriften werden je nach Ebene in das entsprechende HTML-Tag umgewandelt (`h1`, `h2`, `h3`, ...). Sowohl interne wie exter-

[12]http://www.cs.technion.ac.il/~gabr/resources/code/wikiprep/

ne Verlinkungen als auch Kategorien-Verweise werden durch eine eindeutige ID ersetzt und, durch Leerzeichen getrennt, in eigenen XML-Elementen gespeichert (`categories`, `links`, `urls`). Weiterleitungen werden dabei aufgelöst und durch deren Ziel-Artikel ersetzt.

Zusätzlich werden noch fünf weitere Dateien erstellt, die folgendes beinhalten (vgl. Gabrilovich, 2007):

- Alle Ankertexte in Verbindung mit den jeweiligen Link-Zielen,
- die Kategorien-Hierarchie,
- eine Liste mit verwandten Artikeln, die über Text-Marken wie „See also“, „Further information“ oder „Related topic“ identifiziert wurden,
- die Anzahl aller enthaltenen Artikel für jede Kategorie und
- die Anzahl aller eingehenden Links für jeden Artikel.

Templates werden bei der Verarbeitung komplett entfernt.

Ein großer Nachteil von *WikiPrep* im Vergleich zu unserem System ist, dass die erstellte XML-Datei eigentlich nur ein Zwischenprodukt für die weitere Verarbeitung darstellen kann. Ein selektiver Zugriff auf einen bestimmten Artikel ist uneffizient, weil ohne Index erst die komplette Datei nach dem richtigen Element durchsucht werden muss. Bei einer Dateigröße von 20 GB kann das – je nach Position des Artikels in der Datei – einige Sekunden dauern.

Zudem werden bei der Umwandlung sowohl die zwischensprachlichen Verweise als auch die Templates – und damit die Infoboxen – verworfen. Durch die Tatsache, dass die Links komplett vom Text getrennt werden, gehen Kontext-Informationen, wie sie beispielsweise von Ito u. a. (2008) oder Mihalcea (2007) verwendet werden, verloren.

3.3. *WikiXML*

Auch an der Universität von Amsterdam hat man ein System entwickelt, mit dem sich der Wikitext in einem Datenbank-Dump von Wikipedia in ein leichter zu verarbeitendes XML-Format umwandeln lässt. Dazu hat man den in PHP programmierten Parser, der von Wikipedia genutzt wird, so umgeschrieben, dass er nicht HTML, sondern XML ausgibt. Weil das Skript auf dem Original-Parser beruht, werden alle Wikitext-Markup-Elemente konvertiert, sodass die entstehenden XML-Dateien alle wichtigen Informationen beinhalten.

Allerdings wurde der Original-Parser in Version 1.14 zu einem großen Teil neu programmiert. Da *WikiXML* jedoch auf Version 1.8 beruht, ist nicht sichergestellt, dass das Tool noch aktuelle Datenbank-Dumps verarbeiten kann. Die auf der Projekt-Homepage[13] zur Verfügung gestellten Daten sind bereits mehr als eineinhalb Jahre alt.

3.4. *SW1*

Basierend auf einem mit *WikiXML* verarbeiteten Datenbank-Dump haben Atserias u. a. (2008) ein reich annotiertes Corpus erstellt, den „Semantically Annotated Snapshot of the English Wikipedia“. Dazu haben sie die Daten mit einer Reihe von frei verfügbaren Tools verarbeitet. Ziel ihrer Arbeit ist: „[. . .] to provide easy access to syntactic and semantic annotations for researchers of both NLP and IR communities by building a reference corpus to homogenize experiments and make results comparable“ (Atserias u. a., 2008, S. 2313). Die Verarbeitungsschritte waren im Einzelnen (vgl. Atserias u. a., 2008, S. 2313):

- Löschen aller Weiterleitungsseiten
- Entfernen der XML-Tags und die Aufteilung des Textes in Sätze und Token
- Part-of-Speech-Annotation mit dem *SuperSense Tagger*
- Lemmatisierung mit den Funktionen der WordNet-API
- Syntax-Analyse mit *DeSR*
- Semantische Annotierung mit mehreren semantischen Taggern

Das Ergebnis dieser Verarbeitung wurden in 3000 Dateien gespeichert, die durchschnittlich jeweils 500 Einträge enthalten. Als Speicherformat verwenden die Autoren nicht XML, da Überlappungen vorkommen, deren Auflösung nur dadurch möglich gewesen wäre, dass für jedes Token ein eigenes XML-Element verwendet wird (Yahoo!, 2007). Deshalb verwenden Atserias u. a. (2008) ein proprietäres Text-Format namens „Multitag“, bei dem sich jedes Token in einer eigenen Zeile befindet und dessen Annotierungen, jeweils durch Tabulatoren getrennt, rechts davon (vgl. Atserias u. a., 2008, S. 2314).

Das Corpus enthält zwar eine Fülle an zusätzlichen Annotationen, allerdings wurde bei der Weiterverarbeitung der *WikiXML*-Daten ein Teil der in Wikipedia bereits enthaltenen Daten entfernt: Weiterleitungsseiten, Kategorienzugehörigkeit und zwischensprachliche Verweise.

[13] http://ilps.science.uva.nl/WikiXML/

Da *SW1* auf der *WikiXML*-Ausgabe basiert, ist der Snapshot natürlich auch immer nur genau so aktuell wie die zu Grunde liegenden Daten. Deshalb suchen die Autoren nach besseren Möglichkeiten, die Inhalte der Wikipedia aus dem Wikitext-Format zu extrahieren.

3.5. *WikIDF*

Ein Vergleich mehrerer Algorithmen zur Suche verwandter Terme in Wikipedia von Krizhanovsky zeigte, dass mit einer statistischen Text-Analyse die besten Ergebnisse erzielt werden konnten (vgl. Krizhanovsky, 2008, S. 1). Aus diesem Grund entwickelt Krizhanovsky die *WikIDF* genannte Index-Datenbank. Wie der Name bereits vermuten lässt, basiert *WikIDF* auf der TF-IDF-Formel, mit der sich bestimmen lässt, wie wichtig ein Term in einem Dokument ist. Zur Berechnung werden die Termhäufigkeit *tf* und die invertierte Dokumenthäufigkeit *idf* benötigt, wobei *tf* die Anzahl der Vorkommen des Terms im Dokument ist und *idf* die Anzahl aller Dokumente, die diesen Term enthalten.

Das Datenbank-Design von *WikIDF* ist darauf ausgelegt, diese Werte schnell bestimmen zu können:

- Die *term*-Tabelle enthält alle Terme, deren jeweiligen IDF-Wert und die Anzahl aller Vorkommen des Terms im gesamten Corpus.
- Die *page*-Tabelle enthält Titel und Länge aller Artikel (in Wörtern).
- In der *term_page*-Tabelle befinden sich die TF-Werte aller Terme.

Krizhanovsky hat den größten Teil des Wikitext-Markups – und damit auch die darüber ausgezeichneten Informationen – über reguläre Ausdrücke entfernt. So enthält seine Datenbank keinerlei Verweise (Kategorien, Artikel-Verlinkungen, zwischensprachliche Links) und sowohl Templates als auch alle Tabellen werden gelöscht. Damit ist seine Datenbank zwar gut für direkte Suchanfragen geeignet, aber ein Großteil der unter Abschnitt 2.2 beschriebenen Studien wäre damit nicht möglich.

3.6. *Java Wikipedia Library (JWPL)*

Ein weiteres System zum Zugriff auf die Wikipedia-Daten ist die *Java Wikipedia Library*, entwickelt am *Ubiquitous Knowledge Processing Lab* der TU Darmstadt. Die API ist explizit „designed for mining the rich lexical semantic information“ (Zesch u. a., 2008, S. 1646) in der Wikipedia. Damit die Programmierschnittstelle zur NLP-Forschung eingesetzt werden kann, müssen für Zesch u. a. (2008) die folgenden Anforderungen erfüllt

sein: „[it] has to support a [wide] range of access paths, including iteration over all articles, a query syntax, as well as efficient access to information like links, categories, and redirects“ (Zesch u. a., 2008, S. 1649). Um diese Anforderungen zu erfüllen, werden die in den Artikeltexten implizit enkodierten Informationen in einer Datenbank gespeichert.

Ein `Wikipedia`-Objekt stellt die Verbindung mit dieser Datenbank her und erlaubt über verschiedene Methoden den Zugriff auf alle Artikel oder Kategorien. Ein bestimmter Artikel oder eine bestimmte Kategorie lassen sich über die Angabe des jeweiligen Titels ausgeben. Zusätzlich hat man die Möglichkeit, sich über ein `PageQuery`-Objekt nur die Artikel zurückgeben zu lassen, ...

- ... bei denen die folgenden Werte über oder unter einem bestimmten Grenzwert liegen:
  - Anzahl an Kategorien, in denen sich der Artikel befindet
  - Anzahl an einkommenden Links
  - Anzahl an ausgehenden Links
  - Anzahl an Redirects
  - Anzahl der Token
- ... deren Titel einen regulären Ausdruck erfüllt
- ... die Disambiguierungsseiten sind.

Das dabei jeweils zurückgelieferte `Page`-Objekt ermöglicht direkten Zugriff auf die Kategorien, den reinen Artikeltext, die ein- und ausgehenden Links und die Weiterleitungsseiten, die auf diesen Artikel verweisen. Werden weitere Informationen über den Artikel benötigt, kann man sich das zugehörige `ParsedPage`-Objekt holen, über das man beispielsweise Zugriff hat auf die einzelnen Artikel-Abschnitte, Templates, die Ankertexte der Links oder deren Kontext.

Bei der Erstellung der Datenbank werden zwar Informationen wie die Links oder Kategorien aus dem Wikitext der Artikel extrahiert und gesondert gespeichert. Der Artikeltext selbst befindet sich aber weiterhin mit dem kompletten Markup in der Datenbank. Erst beim Zugriff darauf von der API, wird der Wikitext mit dem selbst-implementierten Parser in eine objektorientierte Struktur geladen. Daraus folgt allerdings, dass der reine Artikeltext erst nach diesem Parsing-Schritt zur Verfügung steht und somit nicht für Abfragen verwendet werden kann. Es ist mit der *JWPL* beispielsweise nicht möglich, sich alle Artikel ausgeben zu lassen, in denen das Wort „house“ vorkommt. Ein weiterer Nachteil der *JWPL* ist, dass ihre Lizenz nur den Einsatz für Forschungszwecke erlaubt (vgl. Zesch u. a., 2008, S. 1650).

3.7. Weitere Arbeiten

In diesem Abschnitt wollen wir kurz einige weitere vergleichbare Arbeiten vorstellen, über die wir entweder nicht genug Informationen finden konnten oder die für unsere Arbeit nicht unmittelbar relevant sind.

- *WikiXRay* ist das von Felipe Ortega geschriebene Tool, das er bei seinen unter Abschnitt 2.2.7 vorgestellten Untersuchungen verwendet hat. Es erlaubt die umfassende Analyse aller Revisionsdaten. Dabei werden aber nicht die Artikeltexte selbst untersucht, sondern nur die Artikel-Metadaten (Häufigkeit von Änderungen, Anzahl der Autoren etc.).

- Schenkel u. a. (2007) beschreiben *YAWN!*, ein weiteres System, mit dem sich Wikitext in ein XML-Format umwandeln lässt. Sie verwenden es, um aus den Daten der Wikipedia ein semantisch annotiertes XML-Corpus zu erstellen. Leider konnten wir außer ihrer Veröffentlichung keine weiteren Informationen finden, sodass wir das System keiner näheren Betrachtung unterziehen konnten.

- Auch *Wiki2TEI*[14] ist ein System zur Konvertierung von Wikitext ins XML-Format. Dazu wurde der Original-Parser so umgeschrieben, dass er statt HTML XML ausgibt. Das besondere an diesem Tool ist, dass das zurückgelieferte XML-Format den Richtlinien der *Text Encoding Initiative* folgt. Das Tool wird leider seit geraumer Zeit nicht mehr weiterentwickelt und leidet an den gleichen Problemen wie das oben vorgestellte *WikiXML*. Es basiert sogar auf einer noch älteren Version des MediaWiki-Parsers. Es besteht wenig Aussicht darauf, dass die Programmierer ihr Tool an die aktuelle Version anpassen, denn in der Dokumentation[15] schreiben sie: „We have no intention of keeping this tool in synch (sic!) with more recent versions of the wiki engine: this should not be a problem as long as the Wiki syntax remains unchanged and does not introduce new features.“

3.8. Zusammenfassung

Die oben vorgestellten Arbeiten lassen sich grundsätzlich in zwei Gruppen einteilen:

- *Ressourcen*, die die Daten eines spezifischen Wikipedia-Dumps enthalten: Dabei besteht nicht die Möglichkeit, die Daten zu aktualisieren.

[14] http://wiki2tei.sourceforge.net/
[15] http://wiki2tei.sourceforge.net/Wiki2TeiHelp.html

- *Systeme*, bei denen der Benutzer die Ressource selbst erstellt und dabei entscheiden kann, auf welchem Datenbestand sie beruht.

Das große Manko an den Arbeiten der ersten Gruppe ist, dass dabei die Vorteile der Aktualität und der weiten Abdeckung der Inhalte verloren gehen. Der aktuelle Datenbestand enthält doppelt so viele Artikel wie *SW1* und sogar vier Mal so viele wie das *Wikipedia XML Corpus*.

Durch die direkte Abrufmöglichkeit der Daten stechen vor allem *JWPL* und *WikIDF* hervor, weil sie die Daten in einer Datenbank speichern. Dadurch ist es sofort nach der Konvertierung des Datenbank-Dumps möglich, auf die Inhalte zuzugreifen. Bei den anderen Arbeiten müssen die Daten für einen effizienten Zugriff zuerst in irgendeiner Weise vom Benutzer indiziert werden. Das *Wikipedia XML Corpus* nimmt in dieser Hinsicht allerdings eine gewisse Sonderrolle ein, weil dessen Anspruch ein anderer ist. Es will keine Ressource sein, die alle möglichen Arten von wissenschaftlichen Untersuchungen erlaubt, da es explizit für die Erforschung von XML-Retrieval-Verfahren erstellt wurde.

Doch auch die Datenbank-basierten Systeme haben ihre Schwächen. So ermöglicht zwar die *Java Wikipedia Libary* den Zugriff auf alle wichtigen Informationen. Aber deren Abfrage- und Suchmöglichkeiten sind stark begrenzt. Im Gegensatz dazu erlaubt *WikIDF* die direkte Suche nach einzelnen Wörtern, allerdings enthält es dafür alle anderen Informationen wie den Hyperlink- oder Kategorien-Graph nicht.

Vergleichen wir die Arbeiten bezüglich der enthaltenen Daten, zeigen sich deutliche Vorteile bei den Systemen, die auf dem Original-Parser basieren. Wie wir in Abschnitt 4.5 sehen werden, ist die Verarbeitung von Wikitext-Markup nicht sehr einfach. Die auf dem Original beruhenden Systeme haben den Vorteil, dass sie die Verarbeitungsroutinen für die einzelnen Markup-Elemente nicht neu schreiben, sondern nur an die XML-Ausgabe anpassen müssen. Vor allem bei *WikiPrep* und *WikIDF* wird ein erheblicher Teil der impliziten Informationen verworfen. Doch die Verwendung eines modifizierten Original-Parsers kann auch von Nachteil sein. Da der darin enthaltene PHP-Code zum größten Teil noch prozedural und nicht objektorientiert programmiert ist, ist er viel schwerer zu warten, wie *WikiXML* und *Wiki2TEI* zeigen.

Besonders der *Semantically Annotated Snapshot (SW1)* sticht bei der Menge der Inhalte heraus, da es die einzige der untersuchten Arbeiten ist, die zusätzliche lexikalische und semantische Annotationsdaten enthält.

Zusammenfassend hat sich gezeigt, dass keines der bestehenden Systeme zugleich direkten, effizienten Zugriff mit einem umfassenden Suchmechanismus sowie vielseitige Einsatzmöglichkeiten durch die enthaltene Informationsmenge bietet. Wir werden nun in den nächsten Abschnitten unser System vorstellen, das beide Eigenschaften in sich

vereint.

4. System zur Corpus-Erstellung

Wir werden im Folgenden das System zur Erstellung eines Corpus basierend auf den Daten der englischen Wikipedia vorstellen. Die Reihenfolge der einzelnen Abschnitte orientiert sich dabei grob am tatsächlichen zeitlichen Ablauf der einzelnen Verarbeitungsschritte.

4.1. Anforderungen

Damit das System die zuvor genannten Merkmale bieten kann, sollte es die folgenden Anforderungen erfüllen:

- *Skalierbarkeit:* Da der Datenbestand der Wikipedia immer größer wird, muss das System leicht skalierbar sein.
- *Universelle Einsetzbarkeit*: Das mit dem System generierte Corpus soll für möglichst viele Untersuchungen geeignet sein. Dementsprechend sollten darin – wenn möglich – alle unter Abschnitt 2.2 beschriebenen Daten enthalten sein.
- *Erweiterbarkeit*: Das Speicherformat des Corpus soll so gestaltet sein, dass sich zusätzliche Daten auch nachträglich leicht hinzufügen lassen. Darunter fallen z. B. lexikalische, syntaktische oder semantische Annotationsdaten.
- *Dynamisch*: Dem Benutzer soll es möglich sein, das Corpus selbst zu erstellen. Ihm soll es dabei frei stehen, welchen Datenbestand er zur Generierung des Corpus verwendet.
- *Exportfunktionalität*: Es gibt eine Vielzahl an unterschiedlichen Formaten, mit denen sich textuelle Daten repräsentieren lassen. Das System soll es dem Benutzer leicht machen, die im Corpus enthaltenen Daten in eines dieser Formate zu exportieren.
- *Information Retrieval*: Barcala u. a. (2005) stellen Anforderungen auf, die ein Corpus-System erfüllen muss, damit es zum *Information Retrieval* verwendet werden kann. Deshalb soll das Corpus zusätzlich u. a. die folgenden Anforderungen erfüllen (Barcala u. a., 2005, S. 95 f.) :

Name	Wikipedia XML Corpus	WikiPrep	WikiXML	SW1	JWPL	WikIDF	Eigenes Corpus
Ressource/System	Ressource	System	System	Ressource	System	System	System
Plattform	-	Perl	PHP	-	Java	Java	C++
Speicherformat	XML	XML	XML	Text	DB	DB	DB
Enthaltene Daten							
Interne Verweise (im Kontext)	ja (ja)	ja (nein)	ja (ja)	ja (ja)	ja (ja)	nein (nein)	ja (ja)
Zwischensprachliche Verweise	ja	nein	ja	nein	ja	nein	ja
Kategorien-Hierarchie	ja	ja	ja	nein	ja	nein	ja
Ankertexte[a]	ja	ja	ja	ja	ja	nein	ja
Infoboxen	ja	nein	ja	nein	ja	nein	nein[b]
Weiterleitungen/ Disambiguierung	ja	ja	ja	nein	ja	nein	ja
Artikeltext	ja	ja	ja	ja	ja	ja	ja
Besonderheiten	als Ressource für XML-Retrieval-Verfahren entwickelt		basiert auf OriginalParser	Semantische Annotation	Zusätzliche API für den Zugriff auf Wiktionary	TFIDF-Datenbank	Satzgrenzen-Erkennung, Tokenisierung, POS-Tagging

Tabelle 2: Vergleich der verschiedenen Zugriffsmethoden

[a] in Kombination mit dem Link
[b] Für nächste Version geplant

- Zählfunktionalität: Das Corpus-System soll beispielsweise die Anzahl aller Dokumente ermitteln können, die eine bestimmte Bedingung erfüllen.
- Zusätzliche Informationen: Das System soll auch Metadaten zu den Ergebnissen ausgeben können. In unserem Fall z. B., zu welchen Kategorien ein Dokument gehört.
- Kontext: Bei der Ausgabe der Ergebnisse soll nicht nur die Fundstelle selbst, sondern auch deren Kontext ausgegeben werden können.
- Platzhaltersuche: Bei der Anfrage sollen auch Platzhalter oder reguläre Ausdrücke möglich sein.
- Sortierung: Es soll möglich sein, die Ergebnisse nach bestimmten Kriterien sortieren zu können.

4.2. Plattform

Das Corpus wird in einer relationalen Datenbank gespeichert (siehe Abschnitt 4.9). Als relationales Datenbank-Management-System (RDBMS) wurde aus den folgenden Gründen *PostgreSQL*[16] gewählt:

- Das Datenbank-System ist *quelloffen* und *frei verfügbar*. Es steht unter der BSD-Lizenz, die keinerlei Einschränkungen in der Benutzung macht.
- In den 15 Jahren seit seinerer Entstehung hat sich *PostgreSQL* einen Namen als *zuverlässige* Datenbank gemacht.
- Das RDBMS läuft auf allen gängigen Plattformen und bietet APIs für viele Programmiersprachen (u. a. C/C++, Java, .NET, Perl, Python, Ruby, Tcl, ODBC, ...).
- *PostgreSQL* erlaubt die Verwendung von sogenannten „Tablespaces“. Darüber ist es möglich, die Tabellen und Indizes einer Datenbank beliebig auf mehrere Festplatten zu verteilen, wodurch die Performanz gesteigert werden kann.
- Es unterstützt Unicode.
- Das RDBMS lässt sich gut skalieren, sowohl was die Menge der Daten[17] als auch die Menge der gleichzeitigen Zugriffe anbelangt. So liegt beispielsweise die maximale Tabellengröße bei 32 TB.

[16] http://www.postgresql.org

[17] Die Projekt-Homepage berichtet von Produktivumgebungen, in denen 4 TB an Daten von dem Datenbank-System verwaltet werden(vgl. PostgreSQL, 2009).

Als Programmiersprache wurde C++ verwendet. Dafür sprachen die folgenden Gründe:

- C++ ist *objektorientiert*.
- In C++ lässt sich sehr *performanter* Code schreiben.
- Die Programmiersprache ist sehr *mächtig* und bietet Programmierkonstrukte, die es in vielen anderen Sprachen nicht gibt.
- Die meisten anderen Programmiersprachen ermöglichen es, in C++ geschriebene Bibliotheken einzubinden. Es gibt sogar ein Tool[18], das die dafür benötigten Schnittstellen für viele Sprachen automatisch generieren kann.
- Es gibt viele gut getestete Programmierbibliotheken, von denen in unserem System u. a. die folgenden verwendet wurden:
  - die *XercesC*-Bibliothek des Apache-Projekts[19] zum Parsen von XML
  - *libpqxx*[20] für den Zugriff auf die PostgreSQL-Datenbank
  - ICU4C des *ICU (International Components for Unicode)*-Projekts[21] für die Verarbeitung von Unicode
  - eine Reihe von Bibliotheken des Boost-Projekts[22] zur Verarbeitung von regulären Ausdrücken, für die asynchrone Netzwerk-Kommunikation, für die Implementierung von Multithreading-Fähigkeit und für die (De-)Serialisierung von C++-Objekten

4.3. Architektur

Da die Artikel-Datensätze unabhängig voneinander verarbeitet und gespeichert werden können, bietet es sich an, die einzelnen Verarbeitungsschritte zu parallelisieren. Wir haben diese deshalb auf drei Programme verteilt, die auf verschiedenen Rechnern laufen können.

[18] http://www.swig.org/
[19] http://xerces.apache.org/xerces-c/
[20] *http://pqxx.org/*
[21] http://site.icu-project.org/
[22] http://www.boost.org/

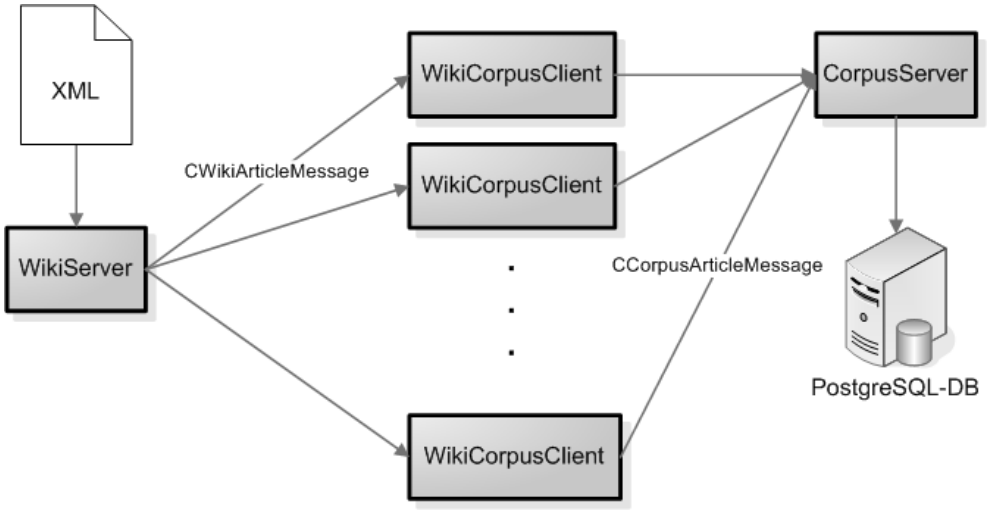

Abbildung 2: Architektur der Komponenten

WikiServer Am *WikiServer* wird die XML-Datei mit allen Artikeln verarbeitet (siehe Abschnitt 4.4). Jedes Mal, wenn ein kompletter Artikel eingelesen ist, wartet der Server auf die Anfrage eines *WikiCorpusClient*-Prozesses. Hat sich ein Client verbunden, wird der Artikel-Datensatz in ein `CWikiArticleMessage`-Objekt serialisiert und übertragen.

Damit der *WikiServer* auf mehrere Anfragen gleichzeitig reagieren kann, laufen mehrere Instanzen des XML-Parsers in verschiedenen Threads.

WikiCorpusClient Der *WikiCorpusClient* arbeitet in einer Endlos-Schleife die folgenden Arbeitsschritte ab:

1. Er verbindet sich zum *WikiServer* und lässt sich einen Artikel übergeben.
2. Das Wikitext-Markup des Artikels wird geparst (siehe Abschnitt 4.5).
3. Aus dem zurückgelieferten Parsebaum werden alle benötigten Inhalte extrahiert (siehe Abschnitt 4.6).
4. Der reine Artikeltext wird dem POS-Tagger zur lexikalischen Verarbeitung übergeben (siehe Abschnitt 4.7).
5. Daraufhin werden alle Term-(Ko-)Okkurrenzen gezählt (siehe Abschnitt 4.8).
6. Alle Informationen werden in einem `CCorpusArticleMessage`-Objekt gespeichert und zum *CorpusServer* übertragen.

Um den Overhead zu verringern, ist es möglich, mehrere verarbeitete Datensätze in einem Puffer zu speichern und gemeinsam zu versenden. Das hat vor allem bei kleineren Artikeln Vorteile, weil dadurch höhere Übertragungsraten erreicht werden können.

CorpusServer Auch der *CorpusServer* läuft in mehreren Threads, um gleichzeitig mit mehreren Clients kommunizieren zu können. Der Server empfängt vom *WikiCorpusClient* einen oder mehrere Artikel-Datensätze und speichert deren Bestandteile in den jeweiligen Datenbank-Tabellen (siehe Abschnitt 4.9).

Client-Server-Kommunikation Zur Netzwerk-Kommunikation wird die `boost::asio`-Bibliothek verwendet, mit der sich asynchrone, nebenläufige Netzwerk-Zugriffe relativ leicht implementieren lassen. Die Logik zum Versenden und Empfangen von Nachrichten befindet sich in der `CConnection`-Klasse, auf die sowohl Clients als auch Server Zugriff haben. Diese Klasse hat entsprechend je zwei Methoden für jede der beiden Nachrichten-Typen: `ReadWikiArticle(..)` und `WriteWikiArticle(..)` für `CWikiArticleMessage`-Objekte und `ReadCorpusArticle(..)` und `WriteCorpusArticle(..)` für `CCorpusArticleMessage`-Objekte. Darin werden die zu übertragenen Daten über die `boost::serialization`-Bibliothek in/aus einen/einem textuellen Datenstrom (de-)serialisiert. Da sich Client und Server nicht notwendigerweise im gleichen Netzwerk befinden müssen und die Client-Server-Kommunikation auch über das Internet stattfinden kann, wurde zusätzlich die Möglichkeit implementiert, die Daten mit der `zlib`-Bibliothek[23] zu komprimieren, um die Bandbreite zu schonen. Die Logik zum (De-)Komprimieren befindet sich in den Methoden `Compress()` und `Decompress()` der Klasse `CDeflate`.

Beim Versand wird vor der eigentlichen Nachricht ein `CMessageHeader`-Objekt übertragen, in dem die folgenden Daten gespeichert sind: der Nachrichten-Typ, die Länge der Nachricht (sowohl komprimiert als auch unkomprimiert) und, ob die Nachricht komprimiert ist oder nicht.

In den folgenden Abschnitten werden nun die einzelnen Verarbeitungsschritte im Detail beschrieben.

4.4. XML-Parser

Wie bereits erwähnt, besteht der Wikipedia-Daten-Dump aus einer einzigen XML-Datei, in der sich alle Artikel befinden. Die Datei hat das folgende Format:

[23] http://www.zlib.net/

```
<mediawiki xmlns="http://www.mediawiki.org/xml/export-0.3/" xmlns:xsi="
    http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="
    http://www.mediawiki.org/xml/export-0.3/ http://www.mediawiki.org/
    xml/export-0.3.xsd" version="0.3" xml:lang="en">
  <siteinfo>
    [ ... Informationen über die Wikipedia-Konfiguration ... ]
  </siteinfo>
  <page>
    <title>Anarchism</title>
    <id>12</id>
    <revision>
      <id>330013878</id>
      <timestamp>2009-12-06T05:56:41Z</timestamp>
      <contributor>
        <username>Mboverload</username>
        <id>49010</id>
      </contributor>
      <comment>[[WP:AWB/T|Typo fixing]], typos fixed: [...]</comment>
      <text xml:space="preserve">{{pp-move-indef}} {{Anarchism sidebar
         }} '''Anarchism''' is a [[political philosophy]] encompassing
          [...]
      </text>
    </revision>
  </page>
  <page>
  [ ... die restlichen Artikel des Wikipedia-Dumps ... ]
</mediawiki>
```

Für jeden Artikel gibt es ein `<page>`-Element, in dessen Sub-Elementen `<title>` und `<text>` sich jeweils Titel und Wikitext des Artikels befinden. Zusätzlich haben jeder Artikel selbst und jede Version davon je eine eindeutige ID.

Für den Zugriff auf XML-Dokumente gibt es zwei verschiedene Schnittstellen: *DOM (Document Object Model)* und *SAX (Simple API for XML)*. Bei der *DOM*-Schnittstelle wird das gesamte Dokument in eine Objektstruktur in den Speicher eingelesen, über die dann auf die Inhalte zugegriffen werden kann. Da der Wikipedia-Dump allerdings sehr groß werden kann – der aktuelle ist größer als 20 GB –, ist der Zugriff über *DOM* nicht geeignet. Wir haben deshalb zum Parsen des XML-Dumps die *SAX2*-Implementierung der *XercesC*-Bibliothek verwendet. Da das Dokument beim *SAX*-Zugriff sequenziell eingelesen wird, können damit auch größere Dateien verarbeitet werden[24]. Der Zugriff auf die Inhalte erfolgt bei *SAX* über sogenannte „Callback-Handler“, deren Methoden bei

[24] In der Standardversion kann XercesC jedoch nur Dateien bis zu 2 GB öffnen. Wir mussten deshalb den Quellcode selbst mit der Compiler-Direktive „LARGEFILE64_SOURCE“ kompilieren.

(im Standard definierten) Ereignissen aufgerufen werden.

Um mit *XercesC* ein XML-Dokument zu parsen, muss über die Methode `createXMLReader()` der `XMLReaderFactory`-Klasse eine Instanz der `SAX2XMLReader`-Klasse erstellt werden. Bei dieser muss man dann den Callback-Handler über die `setContentHandler()`-Methode registrieren.

Die Klasse `CWikiDumpHandler` wurde von der `DefaultHandler`-Klasse abgeleitet und die folgenden Methoden wurden implementiert:

- `startElement()`: Die Methode wird vom XML-Parser aufgerufen, wenn er auf ein öffnendes XML-Tag stößt. In unserer Implementierung wird daraufhin der Name des Elements (`title`, `id`, `text`, `revision` oder `page`) bestimmt und eine Statusvariable entsprechend gesetzt.

- `characters()`: Dieser Methode wird die Zeichenkette zwischen zwei XML-Elementen vom Parser übergeben. Je nach aktuellem Status wird die Zeichenkette der entsprechenden `string`-Variable angehängt.

- `endElement()`: Die Methode wird bei einem schließenden XML-Tag aufgerufen. Handelt es sich dabei um ein `page`-Element, wurde ein kompletter Artikel-Datensatz eingelesen. Damit der XML-Parser Multithreading-fähig ist, wird dem `CWikiDumpHandler`-Objekt bei der Instanziierung übergeben, wieviele Threads parallel laufen. Die Parser der verschiedenen Threads verarbeiten dann jeweils immer nur jeden x-ten Artikel. Ist der Parser für den aktuell eingelesenen Datensatz „zuständig“, wird überprüft, ob es sich um eine Weiterleitungsseite handelt. Ist dies der Fall, wird er sofort in der Datenbank gespeichert. Wenn nicht, werden die Artikel-Daten (Page-ID, Revisions-ID, Titel und Wikitext) in einem `CWikiArticleMessage`-Objekt verpackt und an den nächsten verbundenen *WikiCorpusClient* versendet.

Beim Client erfolgt dann der nächste Verarbeitungsschritt: das Parsen des Wikitext-Markups, das wir im nächsten Abschnitt beschreiben.

4.5. Wikitext-Parser

Der große Erfolg der Wikis kommt u. a. auch daher, dass jeder Artikel schreiben kann, ohne dass er Programmierkenntnisse besitzen müsste. Um Texte zu formatieren, müssen nur einfache Regeln gelernt werden (vgl. Greenstein und Devereux, 2009, S. 6). Diese Regeln werden unter dem Namen „Wikitext(-Markup)“ zusammengefasst. Vom Funktionsumfang her ist Wikitext ähnlich wie HTML, soll aber für den Benutzer leichter zu lesen und zu schreiben sein (vgl. Leuf und Cunningham, 2001).

4.5.1. Wikitext-Markup

Es gibt eigentlich nicht *den* Wikitext, weil jede Wiki-Plattform eine andere Markup-Sprache verwendet[25]. Da die Wikipedia auf der MediaWiki-Software aufbaut, werden wir uns hier auf die Markup-Sprache dieser Plattform beschränken und sie im Folgenden einfach „Wikitext“ nennen.

Die einzelnen Markup-Elemente von Wikitext lassen sich in zwei Gruppen unterteilen: *Unäre* Elemente, die jeweils vor dem auszuzeichnenden Text stehen und *binäre* Elemente, die ihn umklammern (vgl. Mediawiki, 2008).

Unäre Markup-Elemente Über unäre Elemente werden z. B. Listen oder Tabellen ausgezeichnet. Wie man in Abbildung 3 auf der nächsten Seite sehen kann, gibt es drei verschiedene Listen-Typen: Unnummerierte Listen werden durch ein „*“ ausgezeichnet, nummerierte durch ein „#“ und Definitions-Listen durch „;“ (für den zu definierenden Begriff) bzw. „:“ (für die Definitionen). In Wikitext ist es auch möglich, mehrere Listen ineinander zu schachteln. Dabei kann man für jede Listen-Ebene einen anderen Typ verwenden.

Das Tabellen-Markup (siehe Abbildung 3 auf der nächsten Seite) wird mit „{|“ begonnen und mit „|}“ beendet. Die einzelnen Zeilen der Tabelle werden durch „|-“ voneinander getrennt. Die Tabellen-Zellen lassen sich entweder in eine Textzeile schreiben (getrennt durch „||“) oder in mehrere Zeilen (jeweils durch „|“ eingeleitet). Die Tabellen-Überschrift befindet sich in einer eigenen Zeile (eingeleitet mit „|+“).

Der größte Teil der unären Markup-Elemente muss am Zeilenanfang stehen.

Binäre Markup-Elemente Auch in dieser Gruppe gibt es ein Markup-Element, dass sich immer am Zeilenanfang befinden muss: Eine Überschrift wird durch „=“-Zeichen vor und nach dem Überschriften-Text ausgezeichnet und teilt den Artikel automatisch in Abschnitte ein. Die Anzahl der „=“ muss paarweise sein und bestimmt zugleich die Abschnittsebene.

Ebenso werden auch Hyperlinks durch binäres Markup („[[“ und „]]“) ausgezeichnet. Der eingeklammerte Text wird als Titel eines Artikels interpretiert, unabhängig davon, ob es einen Artikel mit diesem Namen gibt oder nicht. Es besteht die Option, einen gesonderten Ankertext einzugeben, der dann statt dem Titel des Link-Ziels im Text

[25] Es gibt zwar ein Projekt, dessen Ziel die Schaffung eines Wikitext-Standards ist (WikiCreole), doch sieht es momentan nicht danach aus, als ob die Wikipedia in der näheren Zukunft auf diesen umstellen würde.

a)

```
== Überschrift Ebene 2 ==
=== Überschrift Ebene 3 ===
```

b)

```
{| <!-- Tabellen-Anfang -->
|+ Tabellen-Überschrift
|- <!-- Anfang der 1. Tabellen-Zeile -->
| 1. Zelle || 2. Zelle || 3. Zelle
|- <!-- Anfang der 2. Tabellen-Zeile -->
| 1. Zelle
| 2. Zelle
| 3. Zelle
|} <!-- Tabellen-Ende -->
```

c)

```
* Erstes Element einer unnummerierten Liste
* Zweites Element einer unnummerierten Liste
*# Erstes Element einer verschachtelten nummerierten Liste
*# Zweites Element einer verschachtelten nummerierten Liste
; Begriffs-Teil einer Definitions-Liste
: Definitions-Teil einer Definitions-Liste
: weitere Definition bei einer Definitions-Liste
```

d)

```
[[Titel des Link-Ziels|Ankertext]]
[[Titel des Link-Ziels, der auch als Ankertext gesetzt wird]]
```

e)

```
{{Infobox President|office=President of the United States|
   term_start=January 20, 2009|vicepresident =[[Joe Biden]]|
   predecessor=[[George W. Bush]]|birthname=Barack Hussein
   Obama II|nationality=[[United States|American]]|party=[[
   Democratic Party (United States)|Democratic]]}}
```

Abbildung 3: Beispiel-Markup für Überschriften (a), Tabellen (b), Listen (c), interne Verweise (d) und Templates (e)

erscheint. Dazu muss, wie in Abbildung 3 auf der vorherigen Seite zu sehen, ein „|“ als Trennung zwischen Link-Ziel und Ankertext eingefügt werden.

Zwischensprachliche Verweise werden nach dem gleichen Prinzip angegeben, indem das entsprechende Sprachkürzel dem Titel des Link-Ziels vorangestellt wird. Auch die Angabe, in welchen Kategorien sich ein Artikel befindet, erfolgt über Verweise (unter Angabe des „Category“-Namensbereichs).

Ein weiterer Bestandteil von Wikitext sind sogenannte Templates. Sie sind im Grunde genommen nichts anderes als Schablonen, deren Inhalt in die Seite kopiert wird. Den Templates lassen sich Parameter übergeben, deren Werte dann an Stellen eingesetzt werden, die im Template-Markup definiert sind. Templates werden in die Seite eingefügt, indem dessen Name in doppelt geschweifte Klammern gesetzt wird. Die Parameter und deren Werte werden, durch „|“ getrennt, direkt hinter den Template-Namen gesetzt (siehe Abbildung 3 auf der vorherigen Seite).

Der Sprachumfang von Wikitext lässt sich über Parser-Erweiterungen vergrößern. Sehr häufig in der Wikipedia verwendet wird z. B. die Erweiterung zum Einfügen von Fußnoten über das `ref`-Tag.

Zur Hervorhebung einzelner Wörter müssen diese von zwei (kursiv), drei (fett) oder fünf (kursiv und fett) Apostrophen umschlossen werden.

Zusätzlich zu den oben beschriebenen Markup-Elementen erlaubt die MediaWiki-Software die Verwendung eines großen Teils des HTML-Markups (z. B. Kommentare, `div`, `span`, `big`, `p`, ...).

Möchte man sicherstellen, dass bestimmte Teile des Artikeltextes nicht in HTML konvertiert werden, kann man sie in `nowiki`-Tags einschließen. Text, der sich darin befindet, wird vom Parser vor der Konvertierung entfernt und erst danach wieder eingefügt.

Funktionsprinzip des Original-Parsers Die Konvertierung beim Original-Parser besteht eigentlich aus einer Reihe von aufeinanderfolgenden Suchen-und-Ersetzen-Vorgängen. Dabei werden für die meisten Markup-Elemente reguläre Ausdrücke verwendet, die das Wikitext-Markup Schritt für Schritt in HTML umwandeln (siehe Abbildung 4 auf Seite 30). Aus diesem Grund ist der Original-Parser auch sehr fehlertolerant.

Um die Problematik bei der Verarbeitung von Wikitext zu verstehen, hilft ein Blick in dessen Entstehungsgeschichte. Der Sprachumfang von Wikitext wurde über die Jahre hinweg immer wieder um neue Konstrukte erweitert (Wikipedia, 2008d). In der Implementierung des Parsers wurde dazu jedes Mal ein neuer Ersetzungsschritt eingefügt. Dabei wurden die Ersetzungsregeln für die einzelnen Markup-Elemente allerdings nie formal definiert. Basierend auf einer formalen Definition ließe sich die Verarbeitung von

Wikitext-Element	Klasse
Tabellen	`CTableElement, CTableCaptionElement, CTableRowElement, CTableCellElement`
Listen	`CListElement, CListItemElement`
Überschriften	`CHeadingElement`
Interne Verweise	`CInternalLinkElement, CInternalImageLinkElement`
Externe Verweise	`CExternalLinkElement`
Templates	`CTemplateElement, CTemplateArgument`
<nowiki>	`CProtectedTextElement`
HTML	`CHtmlTagElement`
Hervorhebungen (fett, kursiv)	`CEmphasizedTextElement`

Tabelle 3: Wikitext-Markup-Elemente und die entsprechenden Klassen

Wikitext stark vereinfachen und beschleunigen. Deshalb gab es in den letzten Jahren immer wieder Versuche, Wikitext mit einer kontextfreien Grammatik zu beschreiben, die bisher allerdings noch nicht von Erfolg gekrönt waren.

Das einzige existierende „Regelwerk“, das die Syntax und Semantik von Wikitext beschreibt, ist die Referenzimplementierung des Original-Parsers. Dieser ist allerdings ein „Monolith,“ bestehend aus mehr als 5000 Zeilen mehr oder weniger prozedural programmierten PHP-Codes.

4.5.2. Implementierung

Einer der wichtigsten Schritte bei der Erstellung des Corpus ist das Parsen des Wikitext-Markups. Um größtmögliche Kompatibilität mit dem Original-Parser zu erreichen, haben wir die dort verwendeten Ersetzungs-Algorithmen übernommen und in einer modular aufgebauten, objekt-orientierten und dadurch leichter zu wartenden C++-Bibliothek implementiert. Dabei haben wir die Algorithmen überall dort verbessert, wo es möglich war, ohne zu riskieren, dass sich der Parser danach anders verhält als das Original.

Anstatt das Markup in HTML zu konvertieren, baut der Parser Schritt für Schritt eine Objekt-Hierarchie auf. Dazu haben wir für (fast) alle Markup-Elemente eigene Klassen implementiert (siehe Tabelle 3), die von der `CWikiElement`-Klasse abgeleitet sind.

Die Hierarchie der `CWikiElement`-Objekte wird in einer Instanz `CWikiElementCollection`-Klasse gespeichert. Zu Beginn des Parsing-Vorgangs enthält dieses Objekt nur eine Instanz der `CWikiTextElement`-Klasse, in der der gesamte Wikitext des Artikels gespeichert ist. Während der einzelnen Ersetzungsvorgänge wird dieses `CWikiTextElement`-Objekt in kleinere Teile aufgetrennt und jeweils einer Instanz des entsprechen-

```
== Center cap ==
A '''center cap''', or '''centercap''' is a decorative
   disk on an [[automobile]] [[wheel]] that covers a
   central portion of the wheel.

== See also ==
* Hubcap
* Car
```

```
<h2>Center cap</h2>
A '''center cap''', or '''centercap''' is a decorative
   disk on an [[automobile]] [[wheel]] that covers a
   central portion of the wheel.

<h2>See also</h2>
* Hubcap
* Car
```

```
<h2>Center cap</h2>
A <b>center cap</b>, or <b>centercap</b> is a decorative
   disk on an [[automobile]] [[wheel]] that covers a
   central portion of the wheel.

<h2>See also</h2>
* Hubcap
* Car
```

```
<h2>Center cap</h2>
A <b>center cap</b>, or <b>centercap</b> is a decorative
   disk on an <a href="/wiki/Automobile" title="
   Automobile">automobile</a> <a href="/wiki/Wheel" title
   ="Wheel">wheel</a> that covers a central portion of
   the wheel.

<h2>See also</h2>
<ul>
<li>Hubcap</li>
<li>Car</li>
</ul>
```

Abbildung 4: Ein Artikeltext in verschiedenen Zwischenstadien der Konvertierung

den Markup-Element-Objekts untergeordnet. So bildet sich über die einzelnen Verarbeitungsschritte ein Baum, dessen Blätter `CWikiTextElement`-Objekte sind. Wir wollen das Funktionsprinzip des Parsers im Folgenden am Beispiel der Verarbeitung von Hervorhebungen weiter verdeutlichen[26].

Das Parsing von Hervorhebungen Die Gültigkeit des Markups für Hervorhebungen ist auf einzelne Zeilen beschränkt. Deshalb erfolgt die Verarbeitung dieses Markups zeilenweise in den folgenden Schritten:

1. Über die `GetText()`-Methode der `CWikiElementCollection` wird der gesamte Artikel-Wikitext geholt und in einer `UnicodeString`-Variable gespeichert.

2. Ein regulärer Ausdruck („^.*?"+.*?$") wird verwendet, um alle Zeilen zu suchen, die mindestens zwei aufeinanderfolgende Apostrophe enthält.

3. In jeder Zeile werden dann die Positionen aller Apostroph-Gruppen ermittelt.

4. Diese werden dann der Reihe nach in einen endlichen Zustandsautomaten eingegeben, der die in Tabelle 4 gezeigte Übergangstabelle verwendet. Dabei werden bei der Aktivierung eines Zustands die jeweiligen `Open...()`- oder `Close...()`-Funktionen aufgerufen. Bei ersteren wird die Startposition in einer Variable gespeichert. Bei letzteren wird eine Instanz der `CEmphasizedTextElement`-Klasse angelegt und der `InsertElement(...)`-Methode der `CWikiElementCollection` zusammen mit Start- und Endposition im Wikitext übergeben.

5. In dieser Methode werden dann die Elemente, die sich momentan in diesem Bereich befinden, extrahiert und der `ProcessElements()`-Methode des `CEmphasizedTextElement`-Objekts übergeben.

6. Diese trennt von den Elementen den Teil ab, in dem sich Start- und End-Markup der Hervorhebung (also die Apostrophe) befinden und speichert den Rest in ihrem eigenen `CWikiElementCollection`-Objekt.

Ein wichtiges Merkmal des Original-Parsers ist, dass er – bedingt durch seinen Aufbau – sehr fehlertolerant ist. Denn fehlerhaftes Markup wird bei der Mustersuche nicht erkannt und dementsprechend auch nicht in HTML konvertiert. Außerdem sind viele Sonderfälle berücksichtigt, sodass der Parser beispielsweise ein fehlendes schließendes Markup bei Hervorhebungen verzeiht und trotzdem die gewünschte Ausgabe liefert.

[26] Tatsächlich ist das Verfahren weitaus komplizierter.

Zustand	2 Apostrophen	3 Apostrophen	5 Apostrophen	Zeilenende

ONLY_B	FIRST_B_THEN_I
	`OpenItalic();`
		NONE
		`CloseBold();`
			ONLY_I
			`CloseBold();`
			`OpenItalic();`
				NONE
				`CloseBold();`

ONLY_I	NONE
	`CloseItalic();`
		FIRST_I_THEN_B
		`OpenBold();`
			ONLY_B
			`CloseItalic();`
			`OpenBold();`
				NONE
				`CloseItalic();`

FIRST_B_THEN_I	ONLY_B
	`CloseItalic();`
		ONLY_I
		`CloseItalic();`
		`CloseBold();`
		`OpenItalic();`
			NONE
			`CloseItalic();`
			`CloseBold();`
				NONE
				`CloseItalic();`
				`CloseBold();`

FIRST_I_THEN_B	ONLY_B
	`CloseBold();`
	`CloseItalic();`
	`OpenBold();`
		ONLY_I
		`CloseBold();`
			NONE
			`CloseBold();`
			`CloseItalic();`
				NONE
				`CloseBold();`
				`CloseItalic();`

BOTH_SIMUL-TANEOUSLY	ONLY_B `CloseItalic();`	ONLY_I `CloseBold();`	NONE `CloseItalic-AndBold();`	NONE `CloseItalic-AndBold();`
NONE	ONLY_I `OpenItalic();`	ONLY_B `OpenBold();`	BOTH_SIMUL-TANEOUSLY `OpenItalicAnd-Bold();`	NONE —

Tabelle 4: Übergangstabelle für das Parsen von Hervorhebungen

Auch unser Parser wurde möglichst fehlertolerant implementiert. So werden Änderungen an der Objekt-Hierarchie wieder zurückgesetzt, wenn dabei Fehler auftraten.

Die aktuelle Version des Parsers kann eingebettetes HTML noch nicht zuverlässig parsen. Eventuell könnte eine leicht veränderte Version von HtmlTidy[27] in den Wikitext-Parser integriert werden, um die Verarbeitung von HTML zu verbessern. Dies bleibt allerdings einer Folgeversion vorbehalten.

Ebenso für die nächste Version ist es geplant, das Parsing von Templates zu erweitern. Da es in der aktuellen Version noch nicht zuverlässig funktioniert, werden Templates momentan zwar geparst, aber für die weitere Verarbeitung aus dem Artikeltext entfernt.

Der Parser ist so gut wie sprachunabhängig, d. h. dass nach kleinen Erweiterungen auch die Inhalte anderer Wikipedia-Versionen damit geparst werden können.

4.6. Datenextraktion

Die Extraktion der benötigten Daten aus dem Parsebaum des Wikitext-Parsers wurde über das *Visitor*-Pattern implementiert. Dazu besitzt die `CWikiElementCollection`-Klasse und jede von `CWikiElement` abgeleitete Klasse die Methode `ProcessContents(..)`. Diese Methode erwartet als Argument ein Objekt einer vom abstrakten Datentyp `CContentProcessor` abgeleiteten Kindklasse. In der `ProcessContents(..)`-Implementierung der `CWikiElementCollection` werden dann jeweils die Implementierungen aller enthaltenen `CWikiElement`-Objekte aufgerufen und das `CContentProcessor`-Objekt übergeben. Die Implementierung der `CContainerElement`-Klasse ruft ihrerseits wieder diese Methode ihrer Unterelemente auf. Dadurch „besucht“ das `CContentProcessor`-Objekt jeden Knoten des Parsebaumes mittels Tiefensuche.

Die `CContentProcessor`-Klasse definiert für den Großteil der von `CWikiElement` abgeleiteten Kindklassen je zwei Methoden `Start...()` und `End...()`, die vom jeweiligen `CWikiElement`-Objekt aufgerufen werden, wenn das `CContentProcessor`-Objekt „zu Besuch“ ist. Damit kann das `CWikiElement`-Objekt dem `CContentProcessor`-Objekt spezifische Informationen über sich übergeben (z. B. die Überschriften-Ebene bei `StartHeading (int level)`). Zusätzlich besitzt die `CContentProcessor`-Klasse eine Methode `AddText(..)`, über die die rein textuellen Inhalte der Wikitext-Elemente übergeben werden können.

Für die Erstellung des Corpus haben wir die konkrete `CCorpusCreator`-Klasse von der abstrakten `CContentProcessor`-Klasse abgeleitet.

Die Inhalte mancher Wikitext-Markup-Elemente, wie beispielsweise Tabellen-Zeilen oder Listen, sind in vielen Fällen keine ganzen Sätze. Damit die darin enthaltenen Wörter

[27] http://tidy.sourceforge.net/#docs

trotzdem jeweils in eigenen Sätzen landen, enthält die `CCorpusCreator`-Klasse eine Liste mit `CTextChunk`-Objekten. Dieser Liste wird in den folgenden Fällen jeweils ein neues Element hinzugefügt:

- am Ende einer Tabellen-Zelle (`EndTableCell()`)
- am Ende einer Tabellen-Zeile (`EndTableRow()`)
- am Ende einer Tabellen-Überschrift (`EndTableCaption()`)
- am Ende einer Tabelle (`EndTable()`)
- am Anfang und am Ende eines Listen-Elements (`StartListItem()` und `EndListItem()`)
- am Ende einer Liste (`EndList()`)
- am Ende einer Überschrift (`EndHeading()`)

Die `AddText(..)`-Implementierung der `CCorpusCreator`-Klasse hängt den übergebenen Text immer an das zuletzt eingefügte `CTextChunk`-Objekt an. Bei der lexikalischen Verarbeitung werden die einzelnen *Textchunks* dann getrennt voneinander verarbeitet, um sicher zu gehen, dass die Inhalte verschiedener *Textchunks* nicht im gleichen Satz landen.

Zusätzlich gibt es für interne Verweise und Hervorhebungen je zwei weitere Listen, in denen die Start- und Endpositionen dieser Elemente gespeichert werden. Dadurch lassen sich die Token nach der lexikalischen Verarbeitung wieder den Markup-Elementen zuordnen, um für die Token eines Link-Ankertextes die entsprechende Verlinkung zu ermitteln oder bei hervorgehobenen Token die Art der Formatierung (fett, kursiv). Diese Zuordnung wird im Moment nur für interne Verweise und Hervorhebungen gespeichert. Es ist allerdings für die nächste Version geplant, auch zu speichern, welche der Token sich in einer Überschrift befinden.

Beim Start eines internen Links wird die `StartInternalLink(..)`-Methode aufgerufen. Dieser Methode wird vom `CInternalLinkElement`-Objekt übergeben, ob es sich um einen normalen Link, einen Kategorien-Link oder einen *Interlanguage*-Link handelt. Für alle drei Verweis-Typen enthält das `CCorpusCreator`-Objekt jeweils ein Set, in dem die Verweise des Artikels gesondert gespeichert werden.

Hat das `CCorpusCreator`-Objekt alle Knoten des Parsebaums traversiert, erfolgt der nächste Schritt: die lexikalische Verarbeitung.

4.7. Lexikalische Verarbeitung

Zur lexikalischen Verarbeitung wurde der FoxTagger (Fuchs, 2007) verwendet. Da dieser POS-Tagger vom Autor programmiert wurde, war es leicht möglich, Anpassungen am Quelltext[28] vorzunehmen. Zudem hat der Tagger bereits sowohl eine Satzgrenzen-Erkennung als auch die Tokenisierung integriert, sodass für diese notwendigen Verarbeitungsschritte kein eigenes Programm verwendet werden musste.

Die Satzgrenzen-Erkennung von FoxTagger basiert auf der folgenden einfachen Heuristik (vgl. Fuchs, 2007, S. 15):

Ein Satz endet grundsätzlich immer bei einem Satzende-Zeichen (siehe Tabelle 5 auf Seite 37), außer in den folgenden Fällen:

- Wenn das Token vor dem Satzende-Zeichen eine Abkürzung ist, die im Lexikon eingetragen ist.
- Wenn das auf das Satzende-Zeichen folgende Wort *nicht* groß geschrieben ist.
- Wenn das Token vor dem Satzende-Zeichen ein einzelner Buchstabe ist.
- Wenn direkt auf das Satzende-Zeichen Anführungsstriche folgen, wird die Satzgrenze hinter jene verschoben.

Als Lexikon verwendet der Tagger nicht – wie sonst üblich – ein aus dem Trainingscorpus erstelltes Lexikon, sondern die Index-Dateien von *WordNet* für die offenen Wortklassen und eine selbst kompilierte Liste für die geschlossenen Wortklassen. Untersuchungen des Autors haben ergeben, dass diese Konfiguration einen positiven Effekt auf die Genauigkeit und die Robustheit des Taggers hat (vgl. Fuchs, 2007, S. 32 ff. und S. 52 ff.). Wir gehen deshalb davon aus, dass der Tagger auch auf dem Datenbestand der Wikipedia sehr gute Ergebnisse liefert, obwohl sich die darin enthaltenen Texte vermutlich stark vom Trainingscorpus unterscheiden. Da es aber leider keinen Gold-Standard für die POS-Annotation der Wikipedia-Texte gibt, konnten wir diese Vermutung nicht verifizieren.

In FoxTagger ist auch eine morphologische Analyse implementiert, die für jedes Token alle für die zugewiesene Wortart möglichen Lemmata ausgibt.

Zur Disambiguierung potenzieller Tags für ein Token sind in FoxTagger verschiedene Algorithmen implementiert, die auf einem zweischichtigen neuronalen Netz und Trigramm-Wahrscheinlichkeiten basieren. Dabei ist der eine Teil der Algorithmen auf

[28] Damit die im vorangegangenen Abschnitt beschriebene Zuordnung zu Markup-Elementen funktioniert, musste der Tagger dahingehend verändert werden, dass er zu jedem Token die Position im Original-Text ausgibt.

Genauigkeit optimiert und der andere Teil auf Performanz. In unserem System haben wir den *PerceptronTagger* (Kontextfenster: vier Token links und drei Token rechts) verwendet, weil dieser den besten Kompromiss zwischen den beiden Größen bietet.

Für das Training des Taggers wurde ein Teil des Brown-Corpus (Francis und Kucera, 1964) verwendet, der an das FoxTagger-Tagset (siehe Tabelle 5 auf der nächsten Seite) angepasst wurde. Der andere Teil des Corpus wurde zur Evaluation verwendet. Dabei erreichte der ausgewählte Disambiguierungs-Algorithmus eine Gesamtgenauigkeit von 96,16 % (vgl. Fuchs, 2007, S. 31).

Wie man in Tabelle 5 auf der nächsten Seite sehen kann, bietet das Tagset von FoxTagger eine klare Trennung von Inhalts- und Funktionswörtern. Diese Funktionalität haben wir unter anderem in der Kookkurrenz-Analyse im nächsten Abschnitt verwendet. Außerdem haben wir die Gesamtzahl von Funktions- und Inhaltswörtern für jeden Artikel bestimmt und im Corpus gespeichert.

Der Tagger gibt zusätzlich zum Tag auch eine Maßzahl aus, wie sicher er sich bei dieser Zuweisung ist (*Certainty*) (vgl. Fuchs, 2007, S. 50). Da sich darüber Tagging-Fehler finden lassen, wird diese Größe für jedes Token im Corpus gespeichert (siehe Abschnitt 4.9).

FoxTagger ist in C# .NET programmiert. Damit er trotzdem von unserem in C++ programmierten System aus aufgerufen werden kann, haben wir ihn mit Mono, einer freien, plattformunabhängigen Implementierung des .NET-Frameworks, neu kompiliert. Denn Mono erlaubt es – im Gegensatz zum .NET-Framework selbst –, .NET-Bibliotheken auch von C++ aus aufzurufen. Dazu haben wir eine eigene C++-Bibliothek geschrieben, die die benötigten Funktionen des Taggers in Wrapper-Klassen kapselt. Die wichtigsten davon sind:

- `CFoxTagger`: Die `TagText(..)`-Methode dieser Klasse bildet den Einstiegspunkt für einen Tagging-Vorgang und liefert dessen Ergebnis einem `CTaggingResult`-Objekt zurück.

- `CTaggingResult`: Die Klasse erlaubt den Zugriff auf die einzelnen Sätze. Zudem gibt es zwei Methoden `GetFunctionWordRate()` und `GetContentWordRate()`, die jeweils den prozentualen Anteil von Funktions- und Inhaltswörtern zurückliefern.

- `CTaggedSentence`: Die Klasse erlaubt den Zugriff auf die einzelnen Token eines Satzes.

- `CTaggedToken`: Diese Klasse repräsentiert ein einzelnes Token des Corpus. Sie bietet Zugriff auf die Oberflächenform des Tokens, alle vom Tagger zurückgelieferten

Tag	Beschreibung	Beispiel
AUX	Hilfsverben	do, was, have, should, can, ...
CC	Konjunktionen	and, that, because, ...
DT	Artikel	the, a, any, this, some, ...
EX	existenzielles „there“	there
IN	Adpositionen	to, for, in, ...
JJ	Adjektive	small, high, ...
NN	Nomina	water, stone, house, ...
PRP	Pronomina	all, anybody, one, four, ...
PRP$	Possessivpronomina	my, mine, yours, ...
RB	Adverbien	absolutely, always, ordinarily, ...
TO	Infinitiv-Marker	to
UH	Interjektionen	ciao, welcome, ...
VB	Verben (Grundform)	pay, bring, come, ...
VBD-VBN	Verben (Präteritum, Partizip Perfekt)	known, followed, charged, gave, ...
VBG	Verben (Gerund, Partizip Präsens)	knowing, following, charging, giving, ...
VBZ	Verben (3. Person Singular)	pays, follows, knows, ...
$	Genitiv-Endungen	', 's
.	Satzende-Zeichen	. ! ? : ;
,	Kommata	,
()	Klammern	(){}{}
-	Trennstrich	-
'	Apostroph, Anführungszeichen	' „“
&	Symbole	& $ %

Tabelle 5: Das Tagset von FoxTagger (vgl. Fuchs, 2007, S. 6 f.)

Lemmata, das zugewiesene Tag, den *Certainty*-Wert, die Position des Tokens im Eingabe-Text, ob das Wort im Lexikon war oder nicht und, ob es sich um ein Inhalts- oder ein Funktionswort handelt.

Nach dem Tagging-Vorgang werden das `CTaggingResult`-Objekt und die im vorherigen Abschnitt erwähnten Sets mit allen Verweisen des Artikels einem `CCorpusArticle`-Objekt übergeben. Dort erfolgt der nächste Schritt: die (Ko-)Okkurrenz-Analyse.

4.8. (Ko-)Okkurrenz-Analyse

Kollokationen sind eines der grundlegenden, aber auch am meisten umstrittenen Konzepte der Corpus-Linguistik. Der dahinter stehende Gedanke ist, dass häufig miteinander auftretende Wörter eine (zunächst nicht näher spezifizierte) Beziehung zueinander haben. Deshalb liegt es nahe, auch das Wikipedia-Corpus auf Kollokationen hin zu untersuchen.

Der nächste Abschnitt wird zunächst den Begriff „Kollokation“ genau definieren und weitere Grundlagen zu diesem Konzept vermitteln. Im darauf folgenden Abschnitt werden wir beschreiben, wie die zur Bestimmung von Kollokationen benötigten Kookkurrenz-Häufigkeiten bei der Erstellung des Corpus ermittelt werden.

4.8.1. Grundlagen

Die Kontroverse um Kollokationen kommt vermutlich zum Teil auch daher, dass der Begriff mehrere verschiedene, wenn auch ähnliche, Bedeutungen hat, die oft miteinander verwechselt werden. Es ist deshalb zweckmäßig, den Begriff zunächst klar zu definieren. Evert grenzt die folgenden Bedeutungen voneinander ab (vgl. Evert, 2008, S. 1213 f.):

- theoretisch: „[L]exicalised, idiosyncratic multiword expressions“ (Evert, 2008, S. 1214) zeichnen sich durch eine fehlende oder zumindest beschränkte Kompositionalität, Ersetzbarkeit und Modifizierbarkeit (vgl. Manning und Schütze, 1999, S. 184) aus und werden im Folgenden als „Mehrwortterm“ bezeichnet.

- empirisch: Kollokationen im Sinne von „recurrent and predictable word combinations, which are a directly observable property of natural language“ (Evert, 2008, S. 1214). Das bedeutet, dass sie sich direkt aus einem Corpus extrahieren lassen und nicht durch „linguistic tests and speaker intuitions“ (Evert, 2008, S. 1215) wie die Mehrwortterme.

In dieser Arbeit gebrauchen wir den Begriff in der zweiten Bedeutung und übernehmen Everts Definition einer Kollokation als

> „a combination of two words that exhibit a tendency to occur near each other in natural language, i. e. *cooccur*“ (Evert, 2008, S. 1214, Herv. im Original).

Dabei unterscheidet man drei verschiedene Typen von Kookkurrenzen, die sich jeweils in der Zählmethode unterscheiden: direkte (*surface*), textuelle (*textual*) und syntaktische (*syntactical*) Kookkurrenzen.

Von direkter Kookkurrenz spricht man, wenn sich die Wörter in unmittelbarer Nähe befinden, üblicherweise gemessen über ein Kontextfenster von drei bis fünf Token. Als Token können dabei entweder nur Wörter oder alle Token (also z. B. auch Satzzeichen) gezählt werden. Zudem hat man die Wahl, ob das Kontextfenster über Satzgrenzen hinaus verschoben werden kann oder nicht (vgl. Evert, 2008, S. 1221).

Textuell kookkurrent sind zwei Wörter, wenn sie sich in der gleichen textuellen Einheit befinden (z. B. Sätze, Paragraphen oder auch ganze Texte). Das hat den Vorteil, dass die Größe des Kontexts nicht so willkürlich bestimmt ist wie bei den direkten Kookkurrenzen (vgl. Evert, 2008, S. 1222). Allerdings entsteht durch die große Menge an Daten zwangsläufig ein gewisses Rauschen (vgl. Evert, 2008, S. 1223).

Zwei Wörter zählen als syntaktisch kookkurrent, wenn sie in einer syntaktischen Beziehung zueinander auftreten (z. B. ein Nomen und ein Adjektiv, das es näher bestimmt). Allerdings wird dazu eine syntaktische Annotation des Corpus benötigt, die selten in ausreichender Genauigkeit vorhanden ist (vgl. Evert, 2008, S. 1224). Eine Annäherung ist jedoch über die Verwendung von Part-of-Speech-Mustern möglich (z. B. direkte Kookkurrenzen von Adjektiv und Nomen im Englischen) (Evert, 2004, S. 38).

Die auf eine dieser Arten ermittelten Kookkurrenz-Häufigkeiten reichen nicht aus, um die Assoziationsstärke von zwei Wörtern zu bestimmen. Denn vor allem bei sehr häufig vorkommenden Wörtern ist es möglich, dass die Kookkurrenzen durch Zufall entstehen. Deshalb muss für jedes Wort des Corpus auch die Marginal-Häufigkeit ermittelt werden (vgl. Evert, 2008, S. 1224), da darüber berechnet werden kann, wie viele Kookkurrenzen der Wahrscheinlichkeit nach zu erwarten sind.

Sogenannte *Assoziationsmaße* vergleichen dann gezählte (O) und erwartete (E) Kookkurrenz-Häufigkeit und zeigen die Assoziationsstärke der beiden Einzelwörter an (siehe Abschnitt 6). Die berechneten Werte können dann verwendet werden, um alle Wortpaare absteigend zu sortieren und die N (100, 500, 1000 oder 2000) besten als „echte Kollokationen“ in die Akzeptanzmenge aufzunehmen. Eine andere Möglichkeit, diese Menge zu bilden, basiert auf einem festgelegten Grenzwert für jedes Assoziationsmaß. Als „echte Kollokationen“ gelten alle Wortpaare, deren Assoziationswert über diesem Schwellenwert liegt (vgl. Evert, 2008, S. 1217).

Kollokationen bieten sich für viele Anwendungsfälle an. So verwendet beispielsweise Rapp Kookkurenz-Daten aus mehreren Corpora, um „sprachliche Prozesse wie das freie Assoziieren, die Ergänzung von Lückentexten und die Bildung syntaktisch orientierter Wortklassen [zu] simulieren“ (Rapp, 1996, S. 3). Ferber schlägt die Verwendung beim *Information Retrieval* vor. Dort könnten sie beispielsweise zur automatischen Erweiterung von Suchanfragen (*query expansion*) um ähnliche Terme, beim Indizierungsprozess

oder zur Generierung von themenspezifischen assoziativen Thesauri genutzt werden (vgl. Ferber, 2003, S. 228). Heyer u. a. (2001a) setzen Kollokationen zur automatischen Extraktion von semantischen Beziehungen zweier Wörter ein. Einen Überblick über weitere Anwendungsmöglichkeiten bietet Evert (vgl. 2004, S. 22–25).

Die Verfahren zur Bestimmung von Kollokationen haben einen entscheidenden Nachteil: Da bei der Zählung der Kookkurrenzen die verschiedenen Bedeutungen der Wörter nicht berücksichtigt werden, sind diese in den Statistiken normalerweise überlagert. Möchte man die Kollokate einzelner Bedeutungen ermitteln, müsste ein ausreichend großes bedeutungsannotiertes Corpus vorhanden sein. Wie wir in Abschnitt 2.2 bereits gesehen haben, wurden die Texte der Wikipedia von Mihalcea (2007) bereits erfolgreich in ein solches Corpus umgewandelt. Wir werden im nächsten Abschnitt das von uns bei der Erstellung des Corpus verwendete Verfahren zur Kookkurrenz-Analyse beschreiben, das unter anderem auch davon Gebrauch macht.

4.8.2. Implementierung

Nachdem der Artikeltext vom POS-Tagger in Token und Sätze aufgeteilt und dem `CCorpusArticle`-Objekt übergeben wurde, erfolgt nun dort die (Ko-)Okkurrenz-Analyse. Dazu werden von jedem Satz, in dem ein Hyperlink vorkommt, zwei Versionen erstellt. In der einen wird dabei das Link-Ziel als Token eingesetzt und in der anderen der Ankertext, wie er auch einem Leser des Artikels auf der Wikipedia-Seite angezeigt werden würde. Für den Satz „Tamale Stadium is a [[multi-purpose stadium]] in [[Tamale, Ghana|Tamale]]“ ergäbe das die folgenden Tokenisierungen[29]:

- „Tamale|Stadium|is|a|multi-purpose_stadium|in|Tamale,_Ghana“ und
- „Tamale|Stadium|is|a|multi-purpose|stadium|in|Tamale“

Auf diese Weise lassen sich die Kookkurrenzen des normalen, aber auch des bedeutungsannotierten Artikel-Texts zählen. Um die beiden Zählweisen zu unterscheiden, werden wir die erste im Folgenden als „Link-Corpus-Kookkurrenz“ und die zweite als „Standard-Corpus-Kookkurrenz“ bezeichnen.

Da die Texte des Corpus nicht syntaktisch annotiert sind, werden nur die direkten und textuellen Kookkurrenzen (auf Artikelebene) gezählt. Für die Zählung der direkten Kookkurrenzen wird ein asymmetrisches Kontextfenster mit einer Größe von vier Token nach rechts verwendet. Zusätzlich werden die Kookkurrenzen dabei für jeden *Slot* des Kontextfensters getrennt voneinander gezählt. Das hat den Vorteil, dass über Mittelwert

[29] Da für die Link-Ziele kein Part-of-Speech-Tag ermittelt wurde, wird diesem ein eigenes Tag „Link“ zugewiesen.

und Varianz der einzelnen Häufigkeiten bestimmt werden kann, wie flexibel bzw. strikt eine Kollokation ist (vgl. Smadja, 1993; Manning und Schütze, 1999, S. 157 ff.). Dabei werden die Abstände einmal mit und einmal ohne Berücksichtigung von Funktionswörtern ermittelt und beide Werte in die Datenbank eingetragen.

Die Häufigkeitswerte für das Wortpaar und dessen Einzelwörter werden bei der Erstellung des Corpus für jeden Artikel getrennt gespeichert. (Dadurch lassen sich die hier ermittelten Werte auch für die Berechnung der Term-Frequenz (TF) verwenden[30].) Wird die Gesamthäufigkeit für das ganze Corpus benötigt, lässt sich das über die Verwendung von Aggregatfunktionen bei der Datenbank-Abfrage leicht ermitteln.

Um die entstehende Datenmenge zu begrenzen, haben wir Kookkurrenzen mit Funktionswörtern nicht gesondert gespeichert. Sie lassen sich aber weiterhin aus den Corpus-Daten extrahieren, wie wir im nächsten Abschnitt zeigen werden.

4.9. Speicherformat

Aufgrund der Menge von Artikeln in der englischen Wikipedia (mehr als 3 Millionen) ist davon auszugehen, dass ein darauf basierendes Corpus sehr umfangreich wird. Damit Suchanfragen trotz dieser Größe sehr schnell Ergebnisse zurückliefern, ist die Wahl des Speicherformates sehr entscheidend. Zudem betreffen auch einige der unter Abschnitt 4.1 genannten Anforderungen die Art der Datenhaltung.

Die meisten Corpora speichern ihre Inhalte in einer „horizontalen“ Darstellung. Dabei sind die einzelnen Token sequenziell gespeichert. Annotationsdaten wie POS-Tags befinden sich, jeweils durch spezielle Markup-Elemente getrennt, zwischen den Token. Neuere Corpora verwenden ein XML-Format zur Trennung von Inhalt und Annotation. Diese Form der Speicherung ist aber vor allem bei sehr großen Corpora nicht sinnvoll, weil kein effizienter Zugriff möglich ist. Es gibt zwar Verfahren, um XML-Corpora zu indizieren, allerdings erlauben diese nur einfache Suchanfragen (vgl. Davies, 2005, S. 309). Außerdem ist es bei Corpora dieser Art schwer, nachträglich Annotationsdaten hinzuzufügen.

Wir haben uns deshalb für eine relationale Datenbank zur Speicherung des Corpus entschieden. Weiterhin sprachen dafür die folgenden Gründe:

- Indizierungs- und Suchstrategien sind bereits im RDBMS implementiert und über Jahre hinweg erprobt und verbessert worden. Darüber sind auch Mustersuchen möglich.

[30] Auf einen weiteren Vorteil dieser Maßnahme kommen wir in Abschnitt 4.9 zurück.

- Außerdem skaliert das verwendete Datenbank-System sehr gut, so dass auch bei der erwarteten Datenmenge keine Performanz-Probleme zu erwarten sind.
- Durch den modularen Aufbau ist es leicht möglich, in der Zukunft weitere Daten hinzuzufügen (z. B. Synonyme), ohne dass dies Auswirkungen auf die Geschwindigkeit hätte (vgl. Davies, 2003, S. 29). Denn hierzu reicht es, entweder zu einer bestehenden Tabelle einfach eine Spalte mit den entsprechenden Daten hinzuzufügen oder eine neue Tabelle anzulegen und sie über einen *Fremdschlüssel* mit den bereits bestehenden Daten zu verknüpfen.
- Durch die Abfragesprache *SQL* ist eine Vielfalt von Abfragen möglich, wie wir in Abschnitt 5 zeigen werden.
- Da die Daten in sehr strukturierter Form vorliegen, ist es sehr leicht möglich, sie in ein anderes Format, wie beispielsweise XML, zu konvertieren. Somit ist auch ein Datenexport relativ einfach zu bewerkstelligen.
- Für die verwendete Datenbank gibt es eine Vielzahl von Programmierschnittstellen, sodass für den programmatischen Zugriff auf das Corpus viele Möglichkeiten zur Verfügung stehen.

Die Corpus-Daten sind auf mehrere Datenbank-Tabellen verteilt. Deren Layout ist dabei so gewählt, dass ein effizienter Zugriff möglich ist. Dabei wird in Kauf genommen, dass die Datenbank sehr groß werden kann.

Zudem haben wir darauf geachtet, dass die Daten der einzelnen Artikel-Versionen getrennt voneinander gespeichert sind. Dadurch ließen sich auch mehrere Versionen eines Artikels gleichzeitig in der Datenbank speichern, um beispielsweise ein diachrones Corpus zu erstellen.

Die wichtigsten Tabellen werden nun im Folgenden genauer beschrieben.

Tabelle *corpus_tokens* In der *corpus_tokens*-Tabelle befinden sich alle Token des Corpus. Wie man in Tabelle 6 auf der nächsten Seite sehen kann, sind die Token nicht – wie sonst üblich – „horizontal“, sondern „vertikal“ gespeichert. Das heißt, dass sich jedes Token des Corpus zusammen mit seinen Annotationsdaten in einer eigenen Zeile befindet.

Zu den Annotationsdaten gehören die möglichen Grundformen (*lemmata*) des Wortes, dessen Wortart (*part_of_speech*), die Information, ob es sich um ein dem Tagger unbekanntes Wort handelt und der oben bereits erwähnte *certainty*-Wert, der angibt wie sicher sich der Tagger bei der Zuweisung des Part-of-Speech-Tags war.

token_pos	surface	lemmata	part_of_speech	certainty		pos_in_sentence	links_to_article	pos_in_link	is_emphasized
394617	A	a	4	98	...	0		-1	f
394618	center	center	9	92	...	1		-1	t
394619	cap	cap	9	30	...	2		-1	t
394620	,	,	3	100	...	3		-1	f
394621	or	or	2	98	...	4		-1	f
394622	center-cap	center-cap	9	97	...	5		-1	t
⋮									
394627	on	on	7	96	...	10		-1	f
394628	an	an	4	98	...	11		-1	f
394629	automobile	automobile	9	99	...	12	Automobile	0	f
394630	wheel	wheel	9	95	...	13	Wheel	0	f

Tabelle 6: Anfang des „Center_cap“-Artikels

Die Position des Token im Corpus ist auf zwei Arten gespeichert: zum einen über die Kombination aus Revisions-ID (*rev_id*), Satznummer im Artikel (*sentence_id*) und Position im Satz (*pos_in_sentence*) und zum anderen über einen eindeutigen Schlüssel über das gesamte Corpus (*token_pos*).

Außerdem ist in der Tabelle für jedes Token gespeichert, ob es im Artikeltext fett oder kursiv hervorgehoben war (*is_emphasized*). Für Token, die sich im Ankertext eines Links befinden, ist zudem vermerkt, auf welchen Artikel der Link verweist (*links_to_article*). Enthält dieser Ankertext mehrere Token, so wird zusätzlich gespeichert, an welcher Position sich die Token jeweils befinden (*pos_in_link*).

Um sich beispielsweise alle Token des „Center_cap“-Artikels in der gleichen Form wie in Tabelle 6 ausgeben zu lassen, genügt das folgende Statement:

```
SELECT * FROM corpus_tokens WHERE rev_id IN (SELECT rev_id
  FROM corpus_articles WHERE title = 'Center_cap') ORDER BY
   token_pos;
```

Die Inhalte der *corpus_tokens*-Tabelle lassen sich auch dazu verwenden, um zusätzlich zu den bereits extrahierten Kookkurrenz-Statistiken weitere n-Gramm-Häufigkeiten zu extrahieren. Um beispielsweise alle mit „new“ beginnenden Trigramme zu ermitteln, kann das folgende SQL-Statement verwendet werden:

Spaltenname	Datentyp	Beschreibung
token_pos	bigint	Die Position des Tokens im Corpus
rev_id	bigint	Die ID der Artikelrevision, in der sich das Token befindet
surface	varchar(255)	Oberflächenform des Tokens, d. h. so wie es im Corpus vorkommt
lemmata	varchar(255)	Alle für das Wort in Kombination mit dem Part-of-Speech-Tag möglichen Grundformen
part_of_speech	smallint	Fremdschlüssel, der auf das entsprechende Part-of-Speech-Tag in der *part_of_speech_tags*-Tabelle verweist
certainty	smallint	Prozentzahl, die angibt, wie sicher sich der POS-Tagger bei der Zuweisung des Tags war
is_unknown_word	boolean	Boole'sche Variable, die angibt, ob es sich um ein, dem Tagger unbekanntes, Wort handelt
sentence_id	smallint	Nummer des Satzes, in dem das Token vorkommt
pos_in_sentence	smallint	Position des Tokens im Satz
links_to_article	varchar(255)	Titel des Artikels, auf den das Token verweist
pos_in_link	smallint	Position des Tokens im Link
is_emphasized	boolean	Boole'sche Variable, die angibt, ob das Token im Artikel hervorgehoben war
creation_timestamp	timestamp	Zeitstempel der Erstellung des Datensatzes

Tabelle 7: Die Tabelle *corpus_tokens*

```
SELECT w1.surface, w2.surface, w3.surface, COUNT(*) FROM
   corpus_tokens w1 JOIN corpus_tokens w2
ON w1.token_pos + 1 = w2.token_pos JOIN corpus_tokens w3 ON
   w2.token_pos + 1 = w3.token_pos WHERE w1.surface = 'new'
   GROUP BY w1.surface, w2.surface, w3.surface ORDER BY
   COUNT(*) DESC;
```

Auf ähnliche Weise ist es, wie oben erwähnt, auch möglich, nachträglich die Kookkurrenz-Häufigkeiten mit Funktionswörtern zu berechnen.

Tabelle *corpus_dictionary* Diese Tabelle wird nach der Erstellung des Corpus aus den Daten der *corpus_tokens*-Tabelle extrahiert. Dazu wird das folgende Statement verwendet:

```
INSERT INTO corpus_dictionary (surface, lemmata,
   part_of_speech, is_unknown_word, occurrence_count)
SELECT surface, lemmata, part_of_speech, is_unknown_word,
   count(surface) FROM corpus_tokens
WHERE part_of_speech NOT IN (1, 3, 6, 11, 15, 17, 18)
GROUP BY surface, lemmata, part_of_speech, is_unknown_word
ORDER BY surface, part_of_speech;
```

Spaltenname	Datentyp	Beschreibung
surface	varchar(255)	Oberflächenform des Wortes, d. h. so wie es im Corpus vorkommt
lemmata	varchar(255)	Alle für das Wort in Kombination mit dem Part-of-Speech-Tag möglichen Grundformen
part_of_speech	smallint	Fremdschlüssel, der auf das entsprechende Part-of-Speech-Tag in der *part_of_speech_tags*-Tabelle verweist
is_unknown_word	boolean	Boole'sche Variable, die angibt, ob es sich um ein dem Tagger unbekanntes Wort handelt
occurrence_count	integer	Anzahl der Vorkommen der Wortform im gesamten Corpus
creation_timestamp	timestamp	Zeitstempel der Erstellung des Datensatzes

Tabelle 8: Die Tabelle *corpus_dictionary*

Das heißt, diese Tabelle entspricht einem Lexikon, das alle im Corpus vorkommenden Wortformen, deren Lemmata sowie Part-of-Speech-Tags enthält, und die Information, ob es sich um ein (für den POS-Tagger) unbekanntes Wort handelt. Zusätzlich wurde über die `COUNT`-Funktion gezählt, wie oft die jeweilige Wortform im Corpus vorkommt. Über den „`NOT IN`"-Teil des Statements werden alle Token herausgefiltert, bei denen es sich nicht um Wörter handelt (z. B. Satzende-Zeichen, Kommata, ...).

Tabelle *corpus_articles* In dieser Tabelle sind Page-ID, Revisions-ID und Titel aller Artikel gespeichert. Sowohl Page-ID als auch Revisions-ID werden direkt aus dem Wikipedia-Daten-Dump übernommen. Erstere (*page_id*) ist eine eindeutige Identifikationsnummer für den Artikel und letztere (*rev_id*) für dessen aktuellste Version.

Wie man in Tabelle 9 sehen kann, enthält jeder Artikel-Datensatz zusätzlich die Anzahl aller Token, aller Inhaltswörter und aller Funktionswörter. Diese Werte werden sowohl für die Version mit den Ankertexten als auch für die Version mit den Link-Zielen gespeichert (siehe Abschnitt 4.8).

Bei Weiterleitungsseiten wird der Titel des Link-Ziels in der Spalte *redirect_target* gespeichert. So lassen sich beispielsweise alle Weiterleitungen, die auf den Artikel „United_States“ verweisen, über das folgende Statement abfragen:

```
SELECT title FROM corpus_articles WHERE redirect_target = '
   United_States';
```

Spaltenname	Datentyp	Beschreibung
page_id	bigint	Aus den Original-Daten übernommene eindeutige ID für den Artikel
rev_id	bigint	Aus den Original-Daten übernommene eindeutige ID für die aktuelle Artikel-Version
title	varchar(255)	Titel des Artikels
redirect_target	varchar(255)	Titel des Artikels, auf den die Seite weiterleitet (bei Weiterleitungsseiten)
token_count_std_corpus	int	Anzahl der Token im Artikel (Standard-Corpus)
content_token_count_std_corpus	int	Anzahl der Inhaltswörter im Artikel (Standard-Corpus)
function_token_count_std_corpus	int	Anzahl der Funktionswörter im Artikel (Standard-Corpus)
token_count_link_corpus	int	Anzahl der Token im Artikel (Link-Corpus)
content_token_count_link_corpus	int	Anzahl der Inhaltswörter im Artikel (Link-Corpus)
function_token_count_link_corpus	int	Anzahl der Funktionswörter im Artikel (Link-Corpus)
creation_timestamp	timestamp	Zeitstempel der Erstellung des Datensatzes

Tabelle 9: Die Tabelle *corpus_articles*

Tabelle *article_links* Die *article_links*-Tabelle (siehe Tabelle 10) speichert den Hyperlink-Graph. Sie enthält für jeden ausgehenden Link eines Artikels je einen Datensatz. Dabei ist der Quell-Artikel über seine Revisions-ID (*article_rev_id*) und der Ziel-Artikel über seinen Titel (*link_target*) referenziert. Aus der Tabelle lassen sich sowohl alle einkommenden als auch alle ausgehenden Links eines Artikels abfragen, wie die folgenden Abfragen am Beispiel des „Center_cap“ (Radkappe)-Artikels zeigen.

Ausgehende Verweise: Da sich der Titel der Link-Quelle nicht in der Tabelle befindet, muss man erst seine Revisions-ID über einen Sub-Select abfragen. Das Statement, um alle ausgehenden Verweise des „Center_cap“-Artikels zu erhalten, lautet dann:

```
SELECT link_target FROM article_links WHERE article_rev_id
   IN (SELECT rev_id FROM corpus_articles WHERE title = '
   Center_cap');
```

Einkommende Verweise: Bei der Abfrage aller einkommenden Verweise erhält man nur die Revisions-ID. Möchte man den zugehörigen Artikelnamen erhalten, muss man die Revisions-ID über einen `JOIN` auflösen:

```
SELECT title FROM article_links JOIN corpus_articles ON
   article_links.article_rev_id = corpus_articles.rev_id
   WHERE link_target = 'Center_cap';
```

Spaltenname	Datentyp	Beschreibung
article_rev_id	bigint	Revisions-ID der Link-Quelle
link_target	varchar(255)	Titel des Link-Ziels
creation_timestamp	timestamp	Zeitstempel der Erstellung des Datensatzes

Tabelle 10: Die Tabelle *article_links*

Sowohl die *article_rev_id*- als auch die *link_target*-Spalte sind indiziert, um einen schnellen Zugriff zu ermöglichen.

Tabelle *article_categories* In der *article_categories*-Tabelle (siehe Tabelle 11) ist gespeichert, in welchen Kategorien sich ein Artikel befindet. Jede Artikel-Revision wird über ihre eindeutige ID (*article_rev_id*) referenziert und jede Kategorie über ihren Namen. Die Tabelle enthält für jeden Artikel und für jede Kategorie, der dieser zugewiesen ist, je einen Datensatz.

Die Kategorien, in denen sich z. B. der Artikel „Center_cap“ befindet, lassen sich über das folgende SQL-Statement abfragen:

```
SELECT category_title FROM article_categories WHERE
   article_rev_id IN (SELECT rev_id FROM corpus_articles
   WHERE title = 'Center_cap');
```

Um sich z. B. die Revisions-IDs aller Artikel der Kategorie „English_outlaws“ zurückgeben zu lassen, muss man nur nach der *category_title*-Spalte filtern. Möchte man jeweils den Titel des Artikels, so reicht ein `JOIN` mit der *corpus_articles*-Tabelle:

```
SELECT title FROM article_categories JOIN corpus_articles ON
    article_categories.article_rev_id = corpus_articles.
   rev_id WHERE category_title = 'English_outlaws';
```

Für einen effizienten Zugriff wurden sowohl auf der *article_rev_id*- als auch auf der *category_title*-Spalte Indizes angelegt.

Spaltenname	Datentyp	Beschreibung
article_rev_id	bigint	Revisions-ID des Artikels
category_title	varchar(255)	Name der Kategorie
creation_timestamp	timestamp	Zeitstempel der Erstellung des Datensatzes

Tabelle 11: Die Tabelle *article_categories*

Tabelle *article_iw_links* Diese Tabelle (siehe Tabelle 12) enthält alle zwischensprachlichen Verweise. Wie bei den beiden vorherigen Tabellen ist der Artikel über seine Revisions-ID (*article_rev_id*) referenziert. Zusätzlich enthält die Tabelle die Spalte *iw_prefix* für das Sprachkürzel (z. B. „de“ für Artikel aus der deutschen oder „fr“ für Artikel aus der französischen Wikipedia) und die Spalte mit dem Artikelnamen des Linkziels (*link_target*).

Dadurch, dass Sprachkürzel und Artikelname getrennt gespeichert werden, lässt sich gezielt nach Verweisen in eine bestimmte Sprachversion filtern. Für eine Liste aller Artikelnamen und deren deutscher Entsprechungen (sofern vorhanden), reicht das folgende Statement:

```
SELECT title, link_target FROM article_iw_links JOIN
   corpus_articles ON article_iw_links.article_rev_id =
   corpus_articles.rev_id WHERE iw_prefix = 'de';
```

Möchte man die deutsche „Übersetzung“ eines bestimmten Artikels (z. B. „Snow_Leopard“), muss nur die `WHERE`-Klausel um die folgende Bedingung erweitert werden: `[...] WHERE iw_prefix = 'de'AND title = 'Snow_Leopard';`

Tabellen *term_cooccurrence_frequencies* und *corpus_cooccurrences* In der ersten der beiden Tabellen (siehe Abschnitt A auf Seite 79 im Anhang) sind die (Ko-)Okkurrenz-Häufigkeiten aller Inhaltswörter für jeden Artikel gespeichert. Sie enthält für jeden Term des Wortpaares drei Spalten, in denen Oberflächenform (*left_surface* und *right_surface*), POS-Tag (*left_part_of_speech* und *right_part_of_speech*) und Lemmata (*left_lemmata* und *right_lemmata*) gespeichert sind.

Spaltenname	Datentyp	Beschreibung
article_rev_id	bigint	Revisions-ID der Link-Quelle
iw_prefix	varchar	Sprachkürzel für das Link-Ziel („de“, „fr“, „it“, . . .)
link_target	varchar(255)	Titel des Link-Ziel-Artikels
creation_timestamp	timestamp	Zeitstempel der Erstellung des Datensatzes

Tabelle 12: Die Tabelle *article_iw_links*

Zusätzlich gibt es für jeden der vier Slots des Kontextfensters je vier weitere Spalten. Darin sind jeweils die Kookkurrenz-Häufigkeiten im Standard-Corpus und Link-Corpus gespeichert. Für beide Zählungen werden die Token-Abstände mit und ohne Berücksichtigung der Funktionswörter berechnet.

Wie bereits erwähnt, lassen sich aus dieser Tabelle auch die Term-Frequenzen der Artikel abfragen. Um sich beispielsweise eine nach der Frequenz absteigende Liste aller Wörter des „Center_cap“-Artikels ausgeben zu lassen, genügt das folgende Statement:

```
SELECT left_lemmata, SUM(left_count_std_corpus) FROM
  term_cooccurrence_frequencies WHERE rev_id IN (SELECT
  rev_id FROM corpus_articles WHERE title = 'Center_cap')
  GROUP BY left_lemmata;
```

Die *corpus_cooccurrences*-Tabelle wurde nach der Erstellung des Corpus aus der *term_coccurrence_frequencies*-Tabelle extrahiert. Sie enthält die aufsummierten (Ko-)Okkurrenz-Häufigkeiten des gesamten Corpus.

Tabelle *function_term_frequencies* Da die *term_cooccurrence_frequencies*-Tabelle keine Kookkurrenzen mit Funktionswörtern enthält, befinden sich darin auch keine Informationen über deren Okkurrenz-Häufigkeiten. Deshalb werden diese in einer eigenen Tabelle gespeichert, damit trotzdem schnell auf diese Informationen zugegriffen werden kann (d. h. ohne die Datensätze sequenziell aus der *corpus_tokens*-Tabelle einlesen zu müssen).

Wie man in Tabelle 13 sehen kann, hat die *function_term_frequencies*-Tabelle einen ähnlichen Aufbau wie die *term_cooccurrence_frequencies*-Tabelle. Der Unterschied ist der, dass sie jeweils nur eine Spalte für die Term-Informationen (*surface, lemmata, part_of_speech, count_std_corpus, count_link_corpus*) enthält.

Spaltenname	Datentyp	Beschreibung
rev_id	bigint	Revisions-ID des Artikels
surface	varchar(255)	Oberflächenform des Funktionswortes
lemmata	varchar(255)	Grundformen des Funktionswortes
part_of_speech	smallint	Part-of-Speech-Tag
is_unknown_word	boolean	Boole'sche Variable, die angibt, ob es sich um ein unbekanntes Wort handelt
count_std_corpus	smallint	Okkurrenz-Häufigkeit im Artikel (Standard-Corpus)
count_link_corpus	smallint	Okkurrenz-Häufigkeit im Artikel (Link-Corpus)
creation_timestamp	timestamp	Zeitstempel der Erstellung des Datensatzes

Tabelle 13: Die Tabelle *function_term_frequencies*

Server	WikiCorpusClient1 & WikiCorpusClient2
Intel Celeron D 2,8 GHz, 1 GB RAM, 2 x 500 GB HDD, Ubuntu 9.04, PostgreSQL 8.3	Intel Xeon Quadcore 2,5 GHz, 1 GB RAM, Ubuntu 9.04, VMWare

Tabelle 14: Spezifikationen der eingesetzten Rechner

5. Evaluation

Das oben beschriebene System wurde dazu eingesetzt, ein Corpus basierend auf dem Daten-Dump der englischen Wikipedia vom 9. Oktober 2009 zu erstellen. Dieser Abschnitt bietet einen Überblick über diesen Testlauf. Anschließend wird untersucht, ob sich das erstellte Corpus für die unter Abschnitt 2.2 vorgestellten wissenschaftlichen Arbeiten einsetzen lässt.

5.1. Testlauf

Bei der Erstellung des Corpus wurden drei Rechner verwendet (siehe Tabelle 14). *WikiServer*- und *CorpusServer*-Prozesse liefen auf dem gleichen PC mit jeweils fünf Threads. Es wurden zwei *WikiCorpusClient*-Prozesse eingesetzt, die beide auf verschiedenen physikalischen Rechnern in einer virtuellen Maschine liefen.

Die Erstellung des Corpus dauerte ca. 3 Wochen. Bei einer Gesamtzahl von 1.595.409.338

(a)

United_-States	266.928
France	117.664
England	105.172
Germany	95.904
United_-Kingdom	92.543
Canada	89.320
World_-War_II	74.129
Australia	70.955
Japan	69.275
2007	63.332

(b)

Living_people	406.151
Articles_-lacking_sources_(Erik9bot)	146.932
Year_of_birth_-missing_(living-_people)	46.227
English-language_films	16.945
American_films	13.948
American_-film_actors	12.644
American_television_actors	10.381
Black_and_-white_films	10.292
English_-footballers	10.046
Year_of_-birth_missing	9.862

(c)

Französisch	530.109
Deutsch	519.974
Italienisch	406.723
Polnisch	404.020
Niederländisch	377.086
Portugiesisch	345.102
Spanisch	342.529
Japanisch	282.323
Russisch	253.339
Schwedisch	209.444

Tabelle 15: Die zehn häufigsten Verweis-Ziele für (a) interne Links, (b) Kategorien und (c) Sprachen

Token entspricht das einer Geschwindigkeit von ca. 880 Token pro Sekunde.

Das fertige Corpus enthält 6.899.655 Artikel-Datensätze, von denen 3.080.622 keine Weiterleitungen sind. Damit enthalten diese im Durchschnitt ungefähr 517 Token, wovon ca. 280 Inhaltswörter sind und ca. 131 Funktionswörter.

In den Artikeln befinden sich insgesamt 73.518.246 Links zu anderen Artikeln, 10.101.027 Kategorien-Links und 6.790.564 Verweise in andere Sprachversionen. Tabelle 15 zeigt jeweils die zehn häufigsten Verweise für jeden Typ.

Insgesamt belegen alle Daten des Corpus knapp über 480 GB Speicherplatz.

5.2. Anwendungsmöglichkeiten

Wir haben in Abschnitt 4.9 bereits gezeigt, dass alle im Corpus enthaltenen Daten mit sehr geringem Aufwand abgefragt werden können. Mit Ausnahme der wissenschaftlichen Arbeiten, die die komplette Versionsgeschichte verwenden, erfüllt das Corpus alle Anforderungen der unter Abschnitt 2.2 vorgestellten Arbeiten, wie die beiden nachfolgenden

Beispiele verdeutlichen.

Explicit Semantic Analysis (ESA) Gabrilovich und Markovitch (2009) benötigen für ihr ESA-Verfahren die Term- und die inversen Dokument-Häufigkeiten. Am Beispiel des „Center_cap"-Artikels werden wir zeigen, wie diese Daten aus dem Corpus extrahiert werden können.

Die Okkurrenz-Häufigkeiten aller Terme im Artikel „Center_cap" werden über das folgende Statement abgefragt ($count(t_i, d_j)$)[31]:

```
SELECT left_lemmata, SUM(left_count_std_corpus) FROM
   term_cooccurrence_frequencies WHERE rev_id IN (SELECT
   rev_id FROM corpus_articles WHERE title = 'Center_cap')
   GROUP BY left_lemmata;
```

Die *inverse Dokument-Häufigkeit* lässt sich dann für die einzelnen Terme folgendermaßen ermitteln (df_i):

```
SELECT COUNT(rev_id) FROM term_cooccurrence_frequencies
   WHERE left_lemmata = 'center' GROUP BY left_lemmata;
```

Gabrilovich und Markovitch (vgl. 2009, S. 447 f.) berechnen den TF-IDF-Wert dann über

$tf(t_1, d_j) \cdot \log \frac{n}{df_i}$,

wobei $tf(t_i, d_j)$ definiert ist als:

$$tf(t_i, d_j) = \begin{cases} 1 + \log count(t_i, d_j), & count(t_i, d_j) > 0 \\ 0, & sonst \end{cases}.$$

Auf die berechneten Werte wird dann eine Cosinus-Normalisierung angewendet. Die zehn Terme des „Center_cap"-Artikels mit den höchsten ESA-Werten haben wir in Tabelle 16 abgebildet.

Association Thesaurus Ito u. a. (2008) erstellen einen Assoziations-Thesaurus, in dem sie die Link-Kookkurrenz-Häufigkeiten zählen. Abbildung 5 zeigt eine schematische Darstellung des Verfahrens. Die von den Autoren benötigte Liste lässt sich z. B. für den Artikel über Angela Merkel folgendermaßen erstellen:

```
SELECT links_to_article FROM corpus_tokens WHERE rev_id IN (
   SELECT rev_id FROM corpus_articles WHERE title =
   Angela_Merkel') AND pos_in_link = 0 ORDER BY token_pos;
```

[31] Dabei beschränken wir uns auf Inhaltswörter, da Funktionswörter bereits per definitionem nicht zum Inhalt des Artikels beitragen. Grundsätzlich ließen sich aber auch diese ohne Weiteres ermitteln.

#	Term
1	hubcap
2	lugnuts
3	cap
4	lug
5	nut
6	centercap
7	decorative
8	wheel
9	rim
10	truck

Tabelle 16: Die zehn Terme des „Center_cap“-Artikel mit den höchsten ESA-Werten

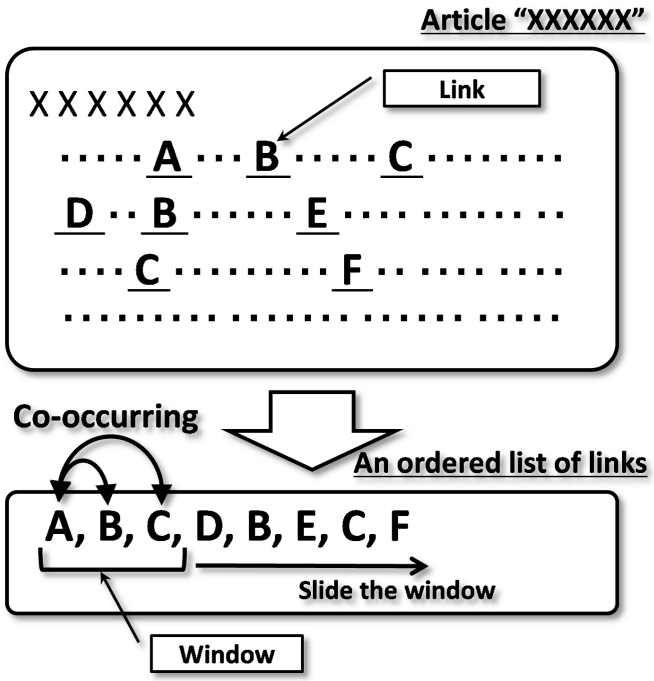

Abbildung 5: Link-Kookkurrenz-Verfahren von Ito u. a. (2008, S. 821)

Weitere Anwendungsmöglichkeiten Darüber hinaus lässt sich das Corpus wegen der Speicherung aller Daten in einer relationalen Datenbank auch zum *Information Retrieval* einsetzen. So könnte ein Benutzer des Corpus nicht nur alle Artikel suchen, in denen das Wort „cut“ vorkommt. Über die Verwendung der Annotationsdaten ließe sich die Suche zusätzlich beispielsweise nur auf Nomen beschränken.

Weiterhin wäre zu untersuchen, ob sich mit dem vorgestellten System auch aus anderen Sprachversionen der Wikipedia Corpora erstellen lassen. Dazu müsste vor allem ein geeigneter POS-Tagger gefunden und in das System integriert werden.

Ebenso wäre auch die Erstellung eines Corpus basierend auf den Diskussions-Seiten interessant, da sich daraus vermutlich wichtige Erkenntnisse über das Diskussions-Verhalten der Wikipedia-Autoren gewinnen ließen.

Eine letzte Erweiterungsmöglichkeit des Systems wurde bereits angesprochen: Die Erstellung eines diachronen Corpus basierend auf den verschiedenen Versionen einzelner Artikel. Dieses könnte dazu verwendet werden, deren Entwicklungsgeschichte auszuwerten.

6. Fazit

Im Folgenden sollen die Ergebnisse der Arbeit zusammengefasst und Verbesserungsvorschläge diskutiert werden.

6.1. Zusammenfassung

Ziel dieser Arbeit war es, die Verwendung der Wikipedia zu Forschungszwecken zu erleichtern. Dazu wurden zuerst einige wissenschaftliche Arbeiten daraufhin analysiert, welche Daten der Enzyklopädie benötigt werden.

Eine Untersuchung vergleichbarer Arbeiten zeigte dann, dass keines der vorgestellten Systeme und Ressourcen sowohl einen direkten und effizienten Zugriff in Kombination mit einer umfassenden Suchfunktion als auch eine universelle Einsetzbarkeit durch die enthaltene Informationsmenge bietet.

Wir stellten ein System vor, dass aus der englischen Wikipedia automatisch ein Textcorpus erstellen kann, das die am häufigsten für die Forschung verwendeten Daten enthält. Zusätzlich wird das Corpus um wertvolle Annotations- und Kookkurrenzdaten angereichert. Als Speicherformat wurde eine relationale Datenbank gewählt. Dadurch lassen sich die dort bereits implementierten Indexierungs- und Such-Funktionen zur Abfrage der Daten verwenden.

Ein Testlauf des Systems und eine anschließende Evaluation zeigten am Beispiel von zwei wissenschaftlichen Arbeiten, dass der Zugriff auf die benötigten Daten mit sehr geringem Aufwand möglich ist. Darüber hinaus haben wir weitere Anwendungsmöglichkeiten vorgestellt.

6.2. Ausblick

Die Nützlichkeit des Corpus ließe sich vor allem durch die Implementierung einer Bedienoberfläche (Corpus-Browser) verbessern, bei der der Benutzer die gewünschten Informationen nicht mehr über SQL-Statements abfragen muss. Dazu bietet sich eine Web-Anwendung an, da sie vom Benutzer keine Installation von Software (außer einem Browser) verlangt und dadurch ein hoher Grad an Plattformunabhängigkeit erreicht werden kann.

Dieser Corpus-Browser soll den für derartige Systeme üblichen Funktionsumfang bieten. Dazu zählt vor allem eine umfrangreiche Suchfunktion, die zusätzlich zu einer einfachen Suche auch die Verwendung von regulären Ausdrücken erlaubt. Dabei soll der Benutzer nicht nur ein einzelnes Wort, sondern auch Phrasen eingeben können. Auch die Annotationsdaten sollen bei der Abfrage miteinbezogen werden können.

Bei der Visualisierung der Suchergebnisse soll vor allem eine *KWIC (Key Word in Context)*-Konkordanz-Ansicht möglich sein. Dabei wird zusätzlich zum abgefragter Suchwort auch der Kontext der Fundstelle mitausgegeben (vgl. Wynne, 2008, S. 710 f.). Auch hier soll der Benutzer entscheiden können, welche der Annotationsdaten mitangezeigt werden (vgl. Wynne, 2008, S. 713). Einzelne Suchergebnisse sollen auf Wunsch erweitert werden können (z. B. auf den gesamten Artikeltext) (vgl. Wynne, 2008, S. 717).

Die Suche soll nicht nur auf dem Corpus selbst, sondern auch auf den daraus extrahierten Kookkurrenzdaten erfolgen können. Auch hier soll es möglich sein, über die Annotationsdaten zu filtern. Zudem sollen die Kollokationen nicht nur als Tabelle, sondern auch in Form eines Graphen dargestellt werden können.

In der gleichen Weise sollen auch die Wikipedia-spezifischen Daten wie die Kategorien-Hierarchie oder der Hyperlink-Graph ausgegeben werden können. Ebenso soll es möglich sein, diese Informationen zum Filtern zu verwenden. Das wäre z. B. auch zum Generieren und Exportieren von Sub-Corpora nützlich.

Weiterhin lässt sich die Einsatzbarkeit des Corpus erweitern, indem auch die in den Seiten vorkommenden Templates gespeichert werden. Dadurch ließen sich z. B. auch die strukturierten Informationen der Infoboxen abfragen.

Die Erstellung des Corpus nimmt mit drei Wochen noch zu viel Zeit in Anspruch. Sollen die Daten des jeweils aktuellsten Wikipedia-Dumps im Corpus-Browser zur Ver-

$$MI = \log_2 \frac{O}{E} \qquad MI^k = \log_2 \frac{O^k}{E} \qquad local - MI = O \cdot \log_2 \frac{O}{E}$$
$$z - score = \frac{O-E}{\sqrt{E}} \qquad t - score = \frac{O-E}{\sqrt{O}} \qquad simple - ll = 2\left(O \cdot \log \frac{O}{E} - (O-E)\right)$$

Abbildung 6: Wichtige einfache Assoziationsmaße (Evert, 2008, S. 1225)

fügung stehen, muss das System diesbezüglich noch optimiert werden. Beim Testlauf zeigte sich, dass vor allem der Server das System ausbremste. Hier könnte durch die Verwendung eines schnelleren Rechners Zeit eingespart werden.

Das Corpus enthält momentan nur die beobachteten Kookkurrenz-Häufigkeiten. Hier wäre es sinnvoll, die Assoziationswerte nicht erst bei Bedarf, sondern direkt nach der Erstellung des Corpus für verschiedene Assoziationsmaße zu berechnen. Evert (vgl. 2008, S. 1229) schlägt dazu vor, erst einfache Assoziationsmaße wie die in Abbildung 6 gezeigten auf die Daten anzuwenden.

Zudem ist geplant auch die IDF-Werte aller Terme in einer eigenen Tabelle zu speichern und gegebenenfalls auch die ESA-Werte in die Datenbank einzutragen.

Auch eine Integration der von der Wikipedia veröffentlichten Benutzungsstatistiken würde den Mehrwert des Corpus vergrößern, da sich daraus Informationen über das Nutzerverhalten ableiten ließen.

Abbildungsverzeichnis

Tabellenverzeichnis

Literatur

[WI2006 2006] : *2006 IEEE / WIC / ACM International Conference on Web Intelligence (WI 2006), 18-22 December 2006, Hong Kong, China*. IEEE Computer Society, 2006. – ISBN 0-7695-2747-7

[AAAI2007 2007] : *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*. AAAI Press, 2007. – ISBN 978-1-57735-323-2

[WI2008 2008] : *2008 IEEE / WIC / ACM International Conference on Web Intelligence, WI 2008, 9-12 December 2008, Sydney, NSW, Australia, Main Conference Proceedings*. IEEE, 2008

[HICSS2008 2008] : *41st Hawaii International International Conference on Systems Science (HICSS-41 2008), Proceedings, 7-10 January 2008, Waikoloa, Big Island, HI, USA*. IEEE Computer Society, 2008

[EACL2009 2009] : *EACL 2009, 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, March 30 - April3, 2009, Athens, Greece*. The Association for Computer Linguistics, 2009

[Adafre und de Rijke 2006] Adafre, Sisay F. ; Rijke, Maarten de: Finding Similar Sentences across Multiple Languages in Wikipedia. In: *Proceedings of the workshop on NEW TEXT Wikis and blogs and other dynamic text sources at EACL 2006*. Trento, Italy, 2006

[Aijmer 2008] Aijmer, Karin: *Parallel and comparable corpora*. Kap. 16, S. 275–292. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[de Alfaro und Ortega 2009] Alfaro, Luca de ; Ortega, Felipe: Measuring Wikipedia: a hands-on tutorial. In: *WikiSym '09: Proceedings of the 5th International Symposium on Wikis and Open Collaboration*. New York, NY, USA : ACM, 2009, S. 1–1. – ISBN 978-1-60558-730-1

[Athenikos und Lin 2009] Athenikos, Sofia J. ; Lin, Xia: Visualizing intellectual connections among philosophers using the hyperlink & semantic data from Wikipedia. In: *WikiSym '09: Proceedings of the 5th International Symposium on Wikis and Open Collaboration*. New York, NY, USA : ACM, 2009, S. 1–2. – ISBN 978-1-60558-730-1

[Atserias u. a. 2008] Atserias, Jordi ; Zaragoza, Hugo ; Ciaramita, Massimiliano ; Attardi, Giuseppe: Semantically Annotated Snapshot of the English Wikipedia. In: (Calzolari u. a., 2008), S. 2313–2316. – http://www.lrec-conf.org/proceedings/lrec2008/. – ISBN 2-9517408-4-0

[Auer und Lehmann 2007] Auer, Sören ; Lehmann, Jens: What Have Innsbruck and Leipzig in Common? Extracting Semantics from Wiki Content. In: Franconi, Enrico (Hrsg.) ; Kifer, Michael (Hrsg.) ; May, Wolfgang (Hrsg.): *Lecture Notes in Computer Science* Bd. 4519. Springer, 2007, S. 503–517. – URL `http://dx.doi.org/10.1007/978-3-540-72667-8_36`

[Auer u. a. 2007] Auer, Sören ; Bizer, Christian ; Kobilarov, Georgi ; Lehmann, Jens ; Ives, Zachary: DBpedia: A Nucleus for a Web of Open Data. In: *In 6th International Semantic Web Conference, Busan, Korea*, Springer, 2007, S. 11–15

[Barcala u. a. 2005] Barcala, Francisco-Mario ; Molinero, Miguel A. ; Domínguez, Eva: Information Retrieval and Large Text Structured Corpora. In: (Moreno-Díaz u. a., 2005), S. 91–100. – ISBN 3-540-29002-8

[Baroni 2008] Baroni, Marco: *Distributions in text.* Kap. 37, S. 803–822. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[Baroni und Evert 2008] Baroni, Marco ; Evert, Stefan: *Statistical methods for corpus exploitation.* Kap. 36, S. 777–803. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[Bechhofer u. a. 2008] Bechhofer, Sean (Hrsg.) ; Hauswirth, Manfred (Hrsg.) ; Hoffmann, Jörg (Hrsg.) ; Koubarakis, Manolis (Hrsg.): *The Semantic Web: Research and Applications, 5th European Semantic Web Conference, ESWC 2008, Tenerife, Canary Islands, Spain, June 1-5, 2008, Proceedings.* Bd. *5021.* Springer, 2008. (Lecture Notes in Computer Science). – ISBN 978-3-540-68233-2

[Bergh und Zanchetta 2008] Bergh, Gunnar ; Zanchetta, Eros: *Web linguistics.* Kap. 18, S. 309–327. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[Biber und Jones 2008] Biber, Douglas ; Jones, James K.: *Quantitative methods in corpus linguistics.* Kap. 61, S. 1287–1305. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[Boutilier 2009] Boutilier, Craig (Hrsg.): *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*. 2009

[Bunescu u. a. 2008] Bunescu, Razvan (Hrsg.) ; Gabrilovich, Evgeniy (Hrsg.) ; Mihalcea, Rada (Hrsg.) ; Association for the Advancement of Artificial Intelligence (Veranst.): *Proceedings of the AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy (WikiAI)*. AAAI Press, July 2008. – AAAI Technical Report WS-08-15

[Bunescu u. a. 2009] Bunescu, Razvan (Hrsg.) ; Gabrilovich, Evgeniy (Hrsg.) ; Mihalcea, Rada (Hrsg.) ; Nastase, Vivi (Hrsg.) ; International Joint Conferences on Artificial Intelligence (Veranst.): *Proceedings of the IJCAI Workshop on User-Contributed Knowledge and Artificial Intelligence: An Evolving Synergy (WikiAI)*. July 2009

[Buriol u. a. 2006] Buriol, Luciana S. ; Castillo, Carlos ; Donato, Debora ; Leonardi, Stefano ; Millozzi, Stefano: Temporal Analysis of the Wikigraph. In: *Web Intelligence*. (WI2006, 2006), S. 45–51. – ISBN 0-7695-2747-7

[Buscaldi und Rosso 2007] Buscaldi, Davide ; Rosso, Paolo: A comparison of methods for the automatic identification of locations in wikipedia. In: *GIR*, 2007, S. 89–92

[Calzolari u. a. 2008] Calzolari, Nicoletta (Hrsg.) ; Choukri, Khalid (Hrsg.) ; Maegaard, Bente (Hrsg.) ; Mariani, Joseph (Hrsg.) ; Odjik, Jan (Hrsg.) ; Piperidis, Stelios (Hrsg.) ; Tapias, Daniel (Hrsg.): *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*. Marrakech, Morocco : European Language Resources Association (ELRA), Mai 2008. – http://www.lrec-conf.org/proceedings/lrec2008/

[Capocci u. a. 2006] Capocci, A. ; Servedio, V. D. P. ; Colaiori, F. ; Buriol, L. S. ; Donato, D. ; Leonardi, S. ; Caldarelli, G.: *Preferential attachment in the growth of social networks: the case of Wikipedia*. Feb 2006. – URL `http://arxiv.org/abs/physics/0602026`

[Carletta u. a. 2005] Carletta, Jean ; Evert, Stefan ; Heid, Ulrich: The NITE XML Toolkit: Data Model and Query Language. In: *Language Resources and Evaluation* 39 (2005), Dezember, Nr. 4, S. 313–334

[Chidlovskii 2009] Chidlovskii, Boris: *Semi-supervised Categorization of Wikipedia Collection by Label Expansion*. Bd. 5631. Kap. 42, S. 412–419. Siehe (Geva u. a., 2009). – ISBN 987-3-642-03760-3

[Christ 1994] Christ, Oliver: A modular and flexible architecture for an integrated corpus query system. In: *In Proceedings of COMPLEX'94*, 1994, S. 7–10

[Dasdan u. a. 2009] Dasdan, Ali ; D'Alberto, Paolo ; Kolay, Santanu ; Drome, Chris: Automatic retrieval of similar content using search engine query interface. In: *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*. New York, NY, USA : ACM, 2009, S. 701–710. – ISBN 978-1-60558-512-3

[Davies 2001] Davies, Mark: *Creating and using multi-million word corpora from web-based newspapers*. S. 58–75. Siehe (Simpson und Swales, 2001)

[Davies 2003] Davies, Mark: *Relational N-Gram Databases as a Basis for Unlimited Annotation on Large Corpora*. 2003

[Davies 2004] Davies, Mark: *Student use of large, annotated corpora to analyze syntactic variation*. Kap. 15, S. 259–269. In: Aston, Guy (Hrsg.) ; Bernardini, Silvia (Hrsg.) ; Stewart, Dominic (Hrsg.): *Corpora and Language Learners*, John Benjamins, 2004

[Davies 2005] Davies, Mark: The advantage of using relational databases for large corpora: Speed, advanced queries, and unlimited annotation. In: *International Journal of Corpus Linguistics* 10 (2005), Nr. 3, S. 307–334

[De Smet und Moens 2009] De Smet, Wim ; Moens, Marie-Francine: Cross-language linking of news stories on the web using interlingual topic modelling. In: *SWSM '09: Proceeding of the 2nd ACM workshop on Social web search and mining*. New York, NY, USA : ACM, 2009, S. 57–64. – ISBN 978-1-60558-806-3

[Denoyer und Gallinari 2006] Denoyer, Ludovic ; Gallinari, Patrick: The Wikipedia XML Corpus. In: (Fuhr u. a., 2007), S. 12–19. – ISBN 978-3-540-73887-9

[Denoyer und Gallinari 2007] Denoyer, Ludovic ; Gallinari, Patrick: Report on the XML mining track at INEX 2005 and INEX 2006: categorization and clustering of XML documents. In: *SIGIR Forum* 41 (2007), Nr. 1, S. 79–90

[Denoyer und Gallinari 2009] Denoyer, Ludovic ; Gallinari, Patrick: *Overview of the INEX 2008 XML Mining Track*. Bd. 5631. Kap. 41, S. 401–411. Siehe (Geva u. a., 2009). – ISBN 987-3-642-03760-3

[Dipper 2008] Dipper, Stefanie: *Theory-driven and corpus-driven computational linguistics, and the use of corpora*. Kap. 5, S. 68–96. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[Dunning 1993] Dunning, Ted: Accurate Methods for the Statistics of Surprise and Coincidence. In: *Computational Linguistics* 19 (1993), Nr. 1, S. 61–74

[Evert 2004] Evert, Stefan: *The Statistics of Word Cooccurrences*, IMS; University of Stuttgart, Dissertation, 2004

[Evert 2008] Evert, Stefan: *Corpora and collocations*. Kap. 58, S. 1212–1249. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[Faulstich u. a. 2005] Faulstich, Lukas C. ; Leser, Ulf ; Lüdeling, Anke: *Storing and Querying Historical Texts in a Relational Database*. 2005 (Informatik-Berichte). – URL `http://edoc.hu-berlin.de/docviews/abstract.php?id=25231`. – [Online: Stand 2009-11-24T13:41:26Z]

[Ferber 2003] Ferber, Reginald: *Information Retrieval. Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web*. Heidelberg : dpunkt.verlag, 2003

[Francis und Kucera 1964] Francis, W. M. ; Kucera, H.: *Brown Corpus. Manual of lnformation*. 1964. – URL `http://khnt.hit.uib.no/icame/manuals/brown/`

[Fuchs 2007] Fuchs, Markus: *Automatische Extraktion und Annotation formaler Textmerkmale*, FH Regensburg, Diplomarbeit, Mai 2007

[Fuhr u. a. 2007] Fuhr, Norbert (Hrsg.) ; Lalmas, Mounia (Hrsg.) ; Trotman, Andrew (Hrsg.): *Comparative Evaluation of XML Information Retrieval Systems, 5th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2006, Dagstuhl Castle, Germany, December 17-20, 2006, Revised and Selected Papers*. Bd. *4518*. Springer, 2007. (Lecture Notes in Computer Science). – ISBN 978-3-540-73887-9

[Gabrilovich 2007] Gabrilovich, Evgeniy: *WikiPrep*. Oktober 2007. – http://www.cs.technion.ac.il/ gabr/resources/code/wikiprep/ (Zugriff am: 08.12.2009)

[Gabrilovich und Markovitch 2007] Gabrilovich, Evgeniy ; Markovitch, Shaul: Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In: *Proceedings of The Twentieth International Joint Conference for Artificial Intelligence*. Hyderabad, India, 2007, S. 1606–1611. – URL `http://www.cs.technion.ac.il/~shaulm/papers/pdf/Gabrilovich-Markovitch-ijcai2007.pdf`

[Gabrilovich und Markovitch 2009] Gabrilovich, Evgeniy ; Markovitch, Shaul: Wikipedia-based Semantic Interpretation for Natural Language Processing. In: *J. Artif. Intell. Res. (JAIR)* 34 (2009), S. 443–498

[Ganjisaffar u. a. 2009] Ganjisaffar, Yasser ; Javanmardi, Sara ; Lopes, Cristina: Leveraging crowdsourcing heuristics to improve search in Wikipedia. In: *WikiSym '09: Proceedings of the 5th International Symposium on Wikis and Open Collaboration*. New York, NY, USA : ACM, 2009, S. 1–2. – ISBN 978-1-60558-730-1

[Gaoying Cui und Chen 2008] Gaoying Cui, Wenjie L. ; Chen, Yirong: Corpus Exploitation from Wikipedia for Ontology Construction. In: (Calzolari u. a., 2008), S. 2125–2132. – http://www.lrec-conf.org/proceedings/lrec2008/. – ISBN 2-9517408-4-0

[Garretson 2008] Garretson, Gregory: Desiderata for Linguistic Software Design. In: *International Journal of English Studies* 8 (2008), Nr. 1, S. 67–94

[Geva u. a. 2009] Geva, Shlomo (Hrsg.) ; Kamps, Jaap (Hrsg.) ; Trotman, Andrew (Hrsg.): *Advances in Focused Retrieval*. Bd. 5631. Springer Berlin / Heidelberg, 2009. – ISBN 987-3-642-03760-3

[Giles 2005] Giles, Jim: Internet encyclopedias go head to head. In: *Nature* 438 (2005), Dezember, S. 900–901. – URL `http://obea.wikispaces.com/file/view/nature_15dec2005_wikipedia.pdf`

[Granitzer u. a. 2009] Granitzer, Michael ; Seifert, Christin ; Zechner, Mario: *Context Based Wikipedia Linking*. Bd. 5631. Kap. 36, S. 354–365. Siehe (Geva u. a., 2009). – ISBN 987-3-642-03760-3

[Greenstein und Devereux 2009] Greenstein, Shane ; Devereux, Michelle: *Wikipedia in the Spotlight*. Juli 2009. – URL `http://www.kellogg.northwestern.edu/faculty/greenstein/images/htm/research/cases/wikipedia.pdf`

[Hammwöhner 2007a] Hammwöhner, Rainer: Interlingual aspects of wikipedia's quality. In: *Proceedings of the 12th International Conference on Information Quality*, 2007

[Hammwöhner 2007b] Hammwöhner, Rainer: Qualitätsaspekte der Wikipedia. In: *kommunikation@gesellschaft* Jg. 8 (2007), Nr. 3. – URL `http://www.soz.uni-frankfurt.de/K.G/B3_2007_Hammwoehner.pdf`

[Han und Zhao 2009] Han, Xianpei ; Zhao, Jun: Named entity disambiguation by leveraging wikipedia semantic knowledge. In: *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*. New York, NY, USA : ACM, 2009, S. 215–224. – ISBN 978-1-60558-512-3

[Happel 2009] Happel, Hans-Jörg: Social search and need-driven knowledge sharing in Wikis with Woogle. In: *WikiSym '09: Proceedings of the 5th International Symposium on Wikis and Open Collaboration*. New York, NY, USA : ACM, 2009, S. 1–10. – ISBN 978-1-60558-730-1

[Heyer u. a. 2001a] Heyer, Gerhard ; Läuter, Martin ; Quasthoff, Uwe ; Wittig, Thomas ; Wolff, Christian: Learning Relations Using Collocations. In: (Maedche u. a., 2001)

[Heyer u. a. 2001b] Heyer, Gerhard ; Läuter, Martin ; Quasthoff, Uwe ; Wolff, Christian: Wissensextraktion durch linguistisches Postprocessing bei der Corpusanalyse. In: (Lobin, 2001), S. 71–83

[Hoey 2008] Hoey, Michael: *Corpus linguistics and word meaning*. Kap. 45, S. 972–987. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[Hoffart u. a. 2009] Hoffart, Johannes ; Zesch, Torsten ; Gurevych, Iryna: An architecture to support intelligent user interfaces for Wikis by means of Natural Language Processing. In: *WikiSym '09: Proceedings of the 5th International Symposium on Wikis and Open Collaboration*. New York, NY, USA : ACM, 2009, S. 1–10. – ISBN 978-1-60558-730-1

[Holloway u. a. 2007] Holloway, Todd ; Bozicevic, Miran ; Börner, Katy: Analyzing and visualizing the semantic coverage of Wikipedia and its authors. In: *Complexity* 12 (2007), Nr. 3, S. 30–40

[Holman Rector 2008] Holman Rector, Lucy: Comparison of Wikipedia and other encyclopedias for accuracy, breadth, and depth in historical articles. In: *Reference Services Review* 36 (2008), Nr. 1, S. 7–22

[Hu u. a. 2007] Hu, Meiqun ; Lim, Ee-Peng ; Sun, Aixin ; Lauw, Hady W. ; Vuong, Ba-Quy: On improving wikipedia search using article quality. In: *WIDM '07: Proceedings of the 9th annual ACM international workshop on Web information and data management*. New York, NY, USA : ACM, 2007, S. 145–152. – ISBN 978-1-59593-829-9

[Huang u. a. 2008] Huang, Anna ; Milne, David ; Frank, Eibe ; Witten, Ian H.: Clustering Documents with Active Learning Using Wikipedia. In: *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*. Washington, DC, USA : IEEE Computer Society, 2008, S. 839–844. – ISBN 978-0-7695-3502-9

[Huang u. a. 2009] Huang, Anna ; Milne, David ; Frank, Eibe ; Witten, Ian H.: Clustering Documents Using a Wikipedia-Based Concept Representation. In: *PAKDD '09: Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*. Berlin, Heidelberg : Springer-Verlag, 2009, S. 628–636. – ISBN 978-3-642-01306-5

[Huang und Croft 2009] Huang, Xuanjing ; Croft, W. B.: A unified relevance model for opinion retrieval. In: *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*. New York, NY, USA : ACM, 2009, S. 947–956. – ISBN 978-1-60558-512-3

[Hundt 2008] Hundt, Marianne: *Text corpora*. Kap. 10, S. 168–187. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[Hunston 2008] Hunston, Susan: *Collection strategies and design decisions*. Kap. 9, S. 154–168. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[Iftene und Balahur-Dobrescu 2008] Iftene, Adrian ; Balahur-Dobrescu, Alexandra: Named Entity Relation Mining using Wikipedia. In: (Calzolari u. a., 2008), S. 763–766. – http://www.lrec-conf.org/proceedings/lrec2008/. – ISBN 2-9517408-4-0

[ILPS 2006] ILPS: *WikiXML*. 2006. – http://ilps.science.uva.nl/WikiXML/ (Zugriff am 08.12.2009)

[Ito u. a. 2008] Ito, Masahiro ; Nakayama, Kotaro ; Hara, Takahiro ; Nishio, Shojiro: Association thesaurus construction methods based on link co-occurrence

analysis for wikipedia. In: (Shanahan u. a., 2008), S. 817–826. – ISBN 978-1-59593-991-3

[de Jong und Kraaij 2006] Jong, F.M.G. de (Hrsg.) ; Kraaij, W. (Hrsg.) ; TNO ICT, Delft, The Netherlands (Veranst.): *6th Dutch-Belgian Information Retrieval Workshop (DIR 2006)*. Neslia Paniculata, Enschede, 2006. – ISBN-10: 90-75296-14-2; ISBN-13: 978-90-75296-14-3

[Kaptein und Kamps 2009] Kaptein, Rianne ; Kamps, Jaap: *Using Links to Classify Wikipedia Pages*. Bd. 5631. Kap. 44, S. 432–435. Siehe (Geva u. a., 2009). – ISBN 987-3-642-03760-3

[Kassner u. a. 2008] Kassner, Laura ; Nastase, Vivi ; Strube, Michael: Acquiring a Taxonomy from the German Wikipedia. In: (Calzolari u. a., 2008), S. 2143–2146. – http://www.lrec-conf.org/proceedings/lrec2008/. – ISBN 2-9517408-4-0

[Kemper u. a. 2007] Kemper, Alfons (Hrsg.) ; Schöning, Harald (Hrsg.) ; Rose, Thomas (Hrsg.) ; Jarke, Matthias (Hrsg.) ; Seidl, Thomas (Hrsg.) ; Quix, Christoph (Hrsg.) ; Brochhaus, Christoph (Hrsg.): *Datenbanksysteme in Business, Technologie und Web (BTW 2007), 12. Fachtagung des GI-Fachbereichs "Datenbanken und Informationssysteme"(DBIS), Proceedings, 7.-9. März 2007, Aachen, Germany*. Bd. *103*. GI, 2007. (LNI). – ISBN 978-3-88579-197-3

[Kimmerle u. a. 2009] Kimmerle, Joachim ; Moskaliuk, Johannes ; Cress, Ulrike: Understanding learning: the Wiki way. In: *WikiSym '09: Proceedings of the 5th International Symposium on Wikis and Open Collaboration*. New York, NY, USA : ACM, 2009, S. 1–8. – ISBN 978-1-60558-730-1

[Kinzler 2008] Kinzler, Daniel: *Automatischer Aufbau eines multilingualen Thesaurus durch Extraktion semantischer und lexikalischer Relationen aus der Wikipedia*, Universität Leipzig, Diplomarbeit, 2008. – URL `http://brightbyte.de/papers/2008/DA/WikiWord.pdf`

[Kittur u. a. 2008] Kittur, Aniket ; Suh, Bongwon ; Chi, Ed H.: Can you ever trust a wiki?: impacting perceived trustworthiness in wikipedia. In: *CSCW*, 2008, S. 477–480

[König 2009] König, René: *WISSENSCHAFT IN WIKIPEDIA UND ANDEREN WIKIMEDIA-PROJEKTEN*, Insitut für Technologiefolgenabschätzung, Österreichische Akademie der Wissenschaften, Mai 2009 (ITA-PROJEKTBERICHT A52-2). – URL `http://epub.oeaw.ac.at/ita/ita-projektberichte/d2-2a52-2.pdf`

[Krenn 2000] Krenn, Brigitte: Empirical Implications on Lexical Association Measures. In: *Proceedings of The Ninth EURALEX International Congress*, 2000

[Krizhanovsky 2008] Krizhanovsky, A. A.: Index wiki database: design and experiments. In: *CoRR* abs/0808.1753 (2008), S. 1–18

[Lüdeling und Kytö 2008] Lüdeling, Anke (Hrsg.) ; Kytö, Merja (Hrsg.): *Corpus Linguistics*. Berlin, New York : Walter de Gruyter, 2008 (Handbooks of Linguistics and Communication Science). – ISBN 978-3-11-021142-9

[Lehmberg und Wörner 2008] Lehmberg, Timm ; Wörner, Kai: *Annotation standards*. Kap. 22, S. 484–501. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[Lehnerer 2006] Lehnerer, Sabine: *Wissensorganisation in der Online-Enzyklopädie Wikipedia*, Universität Regensburg, Diplomarbeit, 2006

[Leuf und Cunningham 2001] Leuf, Bo ; Cunningham, Ward: *The Wiki Way: Collaboration and Sharing on the Internet*. Addison-Wesley Professional, April 2001. – URL `http://www.amazon.ca/exec/obidos/redirect?tag=citeulike04-20{&}path=ASIN/020171499X`

[Lin 1997] Lin, Chin-Yew: *Robust automated topic identification*. Los Angeles, CA, USA, University of Southern California, Dissertation, 1997

[Lobin 2001] Lobin, Henning (Hrsg.): *Proceedings der GLDV-Frühjahrstagung 2001, Sprach- und Textechnologie in digitalen Medien, 28.-30. März 2001 Justus-Liebig-Universität Gießen*. Gesellschaft für linguistische Datenverarbeitung, 2001

[Lowerison und Lowerison 2009] Lowerison, Gretchen ; Lowerison, Michael: Increasing the accuracy of Wiki searches using semantic knowledge engine and semantic archivist. In: *WikiSym '09: Proceedings of the 5th International Symposium on Wikis and Open Collaboration*. New York, NY, USA : ACM, 2009, S. 1–2. – ISBN 978-1-60558-730-1

[Macdonald u. a. 2008] Macdonald, Craig (Hrsg.) ; Ounis, Iadh (Hrsg.) ; Plachouras, Vassilis (Hrsg.) ; Ruthven, Ian (Hrsg.) ; White, Ryen W. (Hrsg.): *Advances in Information Retrieval , 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings*. Bd. *4956*. Springer, 2008. (Lecture Notes in Computer Science). – ISBN 978-3-540-78645-0

[Maedche u. a. 2001] Maedche, Alexander (Hrsg.) ; Staab, Steffen (Hrsg.) ; Nedellec, Claire (Hrsg.) ; Hovy, Eduard H. (Hrsg.): *IJCAI'2001 Workshop on Ontology Learning, Proceedings of the Second Workshop on Ontology Learning OL'2001, Seattle, USA, August 4, 2001 (Held in conjunction with the 17th International Conference on Artificial Intelligence IJCAI'2001)*. Bd. *38*. CEUR-WS.org, 2001. (CEUR Workshop Proceedings)

[Mair 2008] Mair, Christian: *Corpora and the study of recent change in language*. Kap. 52, S. 1109–1126. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[Manning und Schütze 1999] Manning, Christopher D. ; Schütze, Hinrich: *Foundations of statistical natural language processing*. Cambridge, MA, USA : MIT Press, 1999. – ISBN 0-262-13360-1

[Mediawiki 2008] Mediawiki: *Markup spec*. 2008. – URL `http://www.mediawiki.org/wiki/Markup_spec`. – Zugriffsdatum: 12. Dez. 2009

[Mehler 2006] Mehler, Alexander: Text Linkage in the Wiki Medium – a Comparative Study. In: Karlgren, J. (Hrsg.): *Proceedings of the EACL Workshop on New Text – Wikis and Blogs and Other Dynamic Text Sources*. Trento, Italien, April 2006, S. 1–8. – URL `http://www.sics.se/jussi/newtext/working_notes/01_mehler.pdf`

[Mehler 2008] Mehler, Alexander: *Large text networks as an object of corpus linguistic studies*. Kap. 19, S. 328–382. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[Mehler und Gleim 2005] Mehler, Alexander ; Gleim, Rüdiger: The Net for the Graphs - Towards Webgenre Representation for Corpus Linguistic Studies. In: Baroni, Marco (Hrsg.) ; Bernardini, Silvia (Hrsg.): *WaCky! Working papers on the Web as corpus*. Bologna, Italy : Gedit, 2005, S. 191–224. – URL `http://wackybook.sslmit.unibo.it/pdfs/mehler.pdf`. – to appear

[Metaweb Technologies 2009] Metaweb Technologies: *Freebase Wikipedia Extraction (WEX)*. `http://download.freebase.com/wex/`. 2009. – URL `http://download.freebase.com/wex/`

[Mihalcea 2007] Mihalcea, Rada: Using Wikipedia for Automatic Word Sense Disambiguation. In: Sidner, Candace L. (Hrsg.) ; Schultz, Tanja (Hrsg.) ; Zhai, ChengXiang (Hrsg.): *HLT-NAACL*, The Association for Computational Linguistics, 2007, S. 196–203

[Milne 2007] Milne, David: Computing Semantic Relatedness using Wikipedia Link Structure. In: *Proc. of NZCSRSC'07*, 2007

[Milne und Witten 2008] Milne, David ; Witten, Ian H.: Learning to link with wikipedia. In: *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*. New York, NY, USA : ACM, 2008, S. 509–518. – ISBN 978-1-59593-991-3

[Moore 2004] Moore, Robert C.: On Log-Likelihood-Ratios and the Significance of Rare Events. In: Lin, Dekang (Hrsg.) ; Wu, Dekai (Hrsg.): *Proceedings of EMNLP 2004*. Barcelona, Spain : Association for Computational Linguistics, July 2004, S. 333–340

[Moreno-Díaz u. a. 2005] Moreno-Díaz, Roberto (Hrsg.) ; Pichler, Franz (Hrsg.) ; Quesada-Arencibia, Alexis (Hrsg.): *Computer Aided Systems Theory - EUROCAST 2005, 10th International Conference on Computer Aided Systems Theory, Las Palmas de Gran Canaria, Spain, February 7-11, 2005, Revised Selected Papers*. Bd. *3643*. Springer, 2005. (Lecture Notes in Computer Science). – ISBN 3-540-29002-8

[Moturu und Liu 2009] Moturu, Sai T. ; Liu, Huan: Evaluating the trustworthiness of Wikipedia articles through quality and credibility. In: *WikiSym '09: Proceedings of the 5th International Symposium on Wikis and Open Collaboration*. New York, NY, USA : ACM, 2009, S. 1–2. – ISBN 978-1-60558-730-1

[Nothman u. a. 2009] Nothman, Joel ; Murphy, Tara ; Curran, James R.: Analysing Wikipedia and Gold-Standard Corpora for NER Training. In: *EACL*. (EACL2009, 2009), S. 612–620

[Orasan u. a. 2008] Orasan, Constantin ; Hasler, Laura ; Mitkov, Ruslan: *Corpora for text summarisation*. Kap. 60, S. 1271–1287. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[Ortega u. a. 2008] Ortega, Felipe ; González-Barahona, Jesús M. ; Robles, Gregorio: On the Inequality of Contributions to Wikipedia. In: *HICSS*. (HICSS2008, 2008), S. 304

[Ortega 2009] Ortega, JosŽe F.: *Wikipedia: A quantitative analysis*. Madrid, UNIVERSIDAD REY JUAN CARLOS, Dissertation, 2009. – URL `http://libresoft.es/Members/jfelipe/thesis-wkp-quantanalysis`

[Petersen 2004] Petersen, Ulrik: Emdros - a text database engine for analyzed or annotated text. In: *Proceedings of Coling 2004*. Geneva, Switzerland : COLING, Aug 23–Aug 27 2004, S. 1190–1193

[Plummer und Fox 2009] Plummer, Shawn M. ; Fox, Laurie J.: A Wiki: one tool for communication, collaboration, and collection of documentation. In: *SIGUCCS '09: Proceedings of the ACM SIGUCCS fall conference on User services conference*. New York, NY, USA : ACM, 2009, S. 271–274. – ISBN 978-1-60558-477-5

[Ponzetto und Navigli 2009] Ponzetto, Simone P. ; Navigli, Roberto: Large-Scale Taxonomy Mapping for Restructuring and Integrating Wikipedia. In: (Boutilier, 2009), S. 2083–2088

[Ponzetto und Strube 2007] Ponzetto, Simone P. ; Strube, Michael: Deriving a Large-Scale Taxonomy from Wikipedia. In: *AAAI*. (AAAI2007, 2007), S. 1440–1445. – ISBN 978-1-57735-323-2

[PostgreSQL 2009] PostgreSQL: *About PostgreSQL*. 2009. – URL `http://www.postgresql.org/about/`. – Zugriffsdatum: 12. Dez. 2009

[Potamias u. a. 2009] Potamias, Michalis ; Bonchi, Francesco ; Castillo, Carlos ; Gionis, Aristides: Fast shortest path distance estimation in large networks. In: *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*. New York, NY, USA : ACM, 2009, S. 867–876. – ISBN 978-1-60558-512-3

[Potthast u. a. 2008] Potthast, Martin ; Stein, Benno ; Anderka, Maik: A Wikipedia-Based Multilingual Retrieval Model. In: (Macdonald u. a., 2008), S. 522–530. – ISBN 978-3-540-78645-0

[Ramanathan u. a. 2009] Ramanathan, Krishnan ; Sankarasubramaniam, Yogesh ; Mathur, Nidhi ; Gupta, Ajay: Document Summarization using Wikipedia. In: Tiwary, Uma S. (Hrsg.) ; Siddiqui, Tanveer J. (Hrsg.) ; Radhakrishna, M. (Hrsg.) ; Tiwari, M. D. (Hrsg.): *IHCI*, Springer India, 2009, S. 254–260. – URL `http://dblp.uni-trier.de/db/conf/ihci/ihci2009.html#RamanathanSMG09`. – ISBN 978-81-8489-404-2

[Rapp 1996] Rapp, Reinhard ; Hellwig, Peter (Hrsg.) ; Krause, Jürgen (Hrsg.): *Sprache und Computer*. Bd. 16: *Die Berechnung von Assoziationen*. Georg Olms Verlag Hildesheim; Zürich; New York, 1996

[Recchia und Jones 2009] Recchia, Gabriel ; Jones, Michael N.: More data trumps smarter algorithms: Comparing pointwise mutual information with latent semantic analysis. In: *Behavior Research Methods* 41 (2009), Nr. 3, S. 647–656. – URL `http://brm.psychonomic-journals.org/content/41/3/647.abstract`

[Rissanen 2008] Rissanen, Matti: *Corpus linguistics and historical linguistics*. Kap. 4, S. 53–68. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[Romaine 2008] Romaine, Suzanne: *Corpus linguistics and sociolinguistics*. Kap. 6, S. 96–111. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[Roth und Schulte im Walde 2008] Roth, Michael ; Schulte im Walde, Sabine: Corpus Co-Occurrence, Dictionary and Wikipedia Entries as Resources for Semantic Relatedness Information. In: (Calzolari u. a., 2008), S. 1852–1859. – http://www.lrec-conf.org/proceedings/lrec2008/. – ISBN 2-9517408-4-0

[Ruiz-Casado u. a. 2005a] Ruiz-Casado, Maria ; Alfonseca, Enrique ; Castells, Pablo: Automatic Assignment of Wikipedia Encyclopedic Entries to WordNet Synsets. In: (Szczepaniak u. a., 2005), S. 380–386. – ISBN 3-540-26219-9

[Ruiz-Casado u. a. 2005b] Ruiz-Casado, Maria ; Alfonseca, Enrique ; Castells, Pablo: Automatic Extraction of Semantic Relationships for WordNet by Means of Pattern Learning from Wikipedia. URL `http://dx.doi.org/10.1007/11428817_7`, 2005, S. 67–79

[Sangers 2001] Sangers, Larry: *[Nupedia-l] Re: [Advisory-l] The wiki...* 2001. – URL `http://web.archive.org/web/20030414021138/www.nupedia.com/pipermail/nupedia-l/2001-January/000680.html`. – Zugriffsdatum: 12. Dez. 2009

[Schenkel u. a. 2007] Schenkel, Ralf ; Suchanek, Fabian M. ; Kasneci, Gjergji: YAWN: A Semantically Annotated Wikipedia XML Corpus. In: (Kemper u. a., 2007), S. 277–291. – ISBN 978-3-88579-197-3

[Schönhofen 2006] Schönhofen, Peter: Identifying Document Topics Using the Wikipedia Category Network. In: *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*. Washington, DC, USA : IEEE Computer Society, 2006, S. 456–462. – ISBN 0-7695-2747-7

[Schönhofen 2008] Schönhofen, Péter: Annotating Documents by Wikipedia Concepts. In: *Web Intelligence*. (WI2008, 2008), S. 461–467

[Shanahan u. a. 2008] Shanahan, James G. (Hrsg.) ; Amer-Yahia, Sihem (Hrsg.) ; Manolescu, Ioana (Hrsg.) ; Zhang, Yi (Hrsg.) ; Evans, David A. (Hrsg.) ; Kolcz, Aleksander (Hrsg.) ; Choi, Key-Sun (Hrsg.) ; Chowdhury, Abdur (Hrsg.): *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*. ACM, 2008. – ISBN 978-1-59593-991-3

[Shen u. a. 2009] Shen, Dou ; Wu, Jianmin ; Cao, Bin ; Sun, Jian-Tao ; Yang, Qiang ; Chen, Zheng ; Li, Ying: Exploiting term relationship to boost text classification. In: *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*. New York, NY, USA : ACM, 2009, S. 1637–1640. – ISBN 978-1-60558-512-3

[Sigurbjörnsson u. a. 2006] Sigurbjörnsson, Börkur ; Kamps, Jaap ; Rijke, Maarten de: Focused Access to Wikipedia. In: (de Jong und Kraaij, 2006). – ISBN-10: 90-75296-14-2; ISBN-13: 978-90-75296-14-3

[Simpson und Swales 2001] Simpson, Rita C. (Hrsg.) ; Swales, John M. (Hrsg.): *Corpus linguistics in North America*. University of Michigan Press, 2001

[Slashdot 2005] Slashdot: *The Early History of Nupedia and Wikipedia: A Memoir*. 2005. – URL `http://features.slashdot.org/features/05/04/18/164213.shtml`. – Zugriffsdatum: 4. Dez. 2009

[Slavianova 2007] Slavianova, Evguenia: *The LeaP Corpus - Generating a Relational Database for Linguistic Query Support*. Studienarbeit. Januar 2007

[Smadja 1993] Smadja, Frank A.: Retrieving Collocations from Text: Xtract. In: *Computational Linguistics* 19 (1993), Nr. 1, S. 143–177

[Stuckman und Purtilo 2009] Stuckman, Jeff ; Purtilo, James: Measuring the wikisphere. In: *WikiSym '09: Proceedings of the 5th International Symposium on Wikis and Open Collaboration*. New York, NY, USA : ACM, 2009, S. 1–8. – ISBN 978-1-60558-730-1

[Suh u. a. 2009] Suh, Bongwon ; Convertino, Gregorio ; Chi, Ed H. ; Pirolli, Peter: The singularity is not near: slowing growth of Wikipedia. In: *WikiSym '09: Proceedings of the 5th International Symposium on Wikis and Open Collaboration*. New York, NY, USA : ACM, 2009, S. 1–10. – ISBN 978-1-60558-730-1

[Syed u. a. 2008] Syed, Zareen S. ; Finin, Tim ; Joshi, Anupam: Wikipedia as an Ontology for Describing Documents. In: *Proceedings of the 2nd International Conference on Weblogs and Social Media*, AAAI Press, März 2008, S. 136–144

[Szczepaniak u. a. 2005] Szczepaniak, Piotr S. (Hrsg.) ; Kacprzyk, Janusz (Hrsg.) ; Niewiadomski, Adam (Hrsg.): *Advances in Web Intelligence Third International Atlantic Web IntelligenceConference, AWIC 2005, Lodz, Poland, June 6-9, 2005, Proceedings*. Bd. *3528*. Springer, 2005. (Lecture Notes in Computer Science). – ISBN 3-540-26219-9

[Tan u. a. 2009] Tan, Saravadee S. ; Kong, Tang E. ; Sodhy, Gian C.: Annotating wikipedia articles with semantic tags for structured retrieval. In: *SWSM '09: Proceeding of the 2nd ACM workshop on Social web search and mining*. New York, NY, USA : ACM, 2009, S. 17–24. – ISBN 978-1-60558-806-3

[Voss 2005a] Voss, Jakob: *Informetrische Untersuchungen an der Online-Enzyklopädie Wikipedia*, Humboldt-Universität zu Berlin, Diplomarbeit, November 2005. – URL `http://jakobvoss.de/magisterarbeit/MagisterarbeitJakobVoss.pdf`

[Voss 2005b] Voss, Jakob: Measuring Wikipedia. In: Ingwersen, P. (Hrsg.) ; Larsen, B. (Hrsg.): *Proceedings of the 10th International Conference of the International Society for Scientometrics and Informetrics*, Karolinska University Press, 2005, S. 221–231. – URL `http://eprints.rclis.org/archive/00003610/`

[West u. a. 2009] West, Robert ; Precup, Doina ; Pineau, Joelle: Completing wikipedia's hyperlink structure through dimensionality reduction. In: *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*. New York, NY, USA : ACM, 2009, S. 1097–1106. – ISBN 978-1-60558-512-3

[Wikipedia 2008a] Wikipedia: *HTML in wikitext*. 2008. – URL `http://en.wikipedia.org/wiki/HTML_in_wikitext`. – Zugriffsdatum: 12. Dez. 2009

[Wikipedia 2008b] Wikipedia: *Template*. 2008. – URL `http://en.wikipedia.org/wiki/Template`. – Zugriffsdatum: 12. Dez. 2009

[Wikipedia 2008c] Wikipedia: *Wikipedia in academic studies*. 2008. – URL `http://en.wikipedia.org/wiki/Wikipedia:Wikipedia_in_academic_studies`. – Zugriffsdatum: 12. Dez. 2009

[Wikipedia 2008d] Wikipedia: *WikipediaWeekly*. 2008. – URL `http://en.wikipedia.org/wiki/Wikipedia:WikiProject_WikipediaWeekly/CurrentTranscriptions/Episode6#Wiki_text_hard_to_parse.3F`. – Zugriffsdatum: 12. Dez. 2009

[Wikipedia 2008e] Wikipedia: *Wikitext*. 2008. – URL `http://en.wikipedia.org/wiki/Wikitext`. – Zugriffsdatum: 12. Dez. 2009

[Wimalasuriya und Dou 2009] Wimalasuriya, Daya C. ; Dou, Dejing: Using multiple ontologies in information extraction. In: *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*. New York, NY, USA : ACM, 2009, S. 235–244. – ISBN 978-1-60558-512-3

[Wöhner und Peters 2009] Wöhner, Thomas ; Peters, Ralf: Assessing the quality of Wikipedia articles with lifecycle based metrics. In: *WikiSym '09: Proceedings of the 5th International Symposium on Wikis and Open Collaboration*. New York, NY, USA : ACM, 2009, S. 1–10. – ISBN 978-1-60558-730-1

[Wu und Weld 2007] Wu, Fei ; Weld, Daniel S.: Autonomously semantifying wikipedia. In: *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. New York, NY, USA : ACM, 2007, S. 41–50. – URL `http://dx.doi.org/10.1145/1321440.1321449`. – ISBN 978-1-59593-803-9

[Wynne 2008] Wynne, Martin: *Searching and concordancing*. Kap. 33, S. 706–737. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[Xiao 2008] Xiao, Richard: *Well-known and influential corpora*. Kap. 20, S. 383–457. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[Xiong u. a. 2009] Xiong, Yuhong ; Luo, Ping ; Zhao, Yong ; Lin, Fen ; Feng, Shicong ; Zhou, Baoyao ; Zheng, Liwei: OfCourse: web content discovery, classification and information extraction for online course materials. In: *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*. New York, NY, USA : ACM, 2009, S. 2077–2078. – ISBN 978-1-60558-512-3

[Yahoo! 2007] Yahoo!: *Why not XML?* 2007. – URL `http://www.yr-bcn.es/dokuwiki/doku.php?id=semantically_annotated_snapshot_of_wikipedia`. – Zugriffsdatum: 12. Dez. 2009

[Zaragoza u. a. 2007a] Zaragoza, H. ; Atserias, J. ; Ciaramita, M. ; Attardi, G.: *Semantically Annotated Snapshot of the English Wikipedia v.1 (SW1).* `http://www.yr-bcn.es/semanticWikipedia`. 2007

[Zaragoza u. a. 2007b] Zaragoza, Hugo ; Rode, Henning ; Mika, Peter ; Atserias, Jordi ; Ciaramita, Massimiliano ; Attardi, Giuseppe: Ranking very many typed entities on wikipedia. In: *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management.* New York, NY, USA : ACM, 2007, S. 1015–1018. – ISBN 978-1-59593-803-9

[Zesch u. a. 2008] Zesch, Torsten ; Müller, Christof ; Gurevych, Iryna: Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In: (Calzolari u. a., 2008), S. 1646–1652. – http://www.lrec-conf.org/proceedings/lrec2008/. – ISBN 2-9517408-4-0

[Zhang u. a. 2009] Zhang, Yan ; Jiang, Qiancheng ; Zhang, Lei ; Zhu, Yizhen: Exploiting bidirectional links: making spamming detection easier. In: *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management.* New York, NY, USA : ACM, 2009, S. 1839–1842. – ISBN 978-1-60558-512-3

[Zierl 1997] Zierl, Marco: *Entwicklung und Implementierung eines Datenbanksystems zur Speicherung und Verarbeitung von Textkorpora*, Friedrich-Alexander-Universität Erlangen-Nürnberg, Diplomarbeit, 1997. – URL `http://www.linguistik.uni-erlangen.de/files/zierl97.pdf`

[Zinsmeister u. a. 2008] Zinsmeister, Heike ; Hinrichs, Erhard ; Kübler, Sandra ; Witt, Andreas: *Linguistically annotated corpora: Quality assurance, reusability and sustainability.* Kap. 35, S. 759–778. Siehe (Lüdeling und Kytö, 2008). – ISBN 978-3-11-021142-9

[Zirn u. a. 2008] Zirn, Cäcilia ; Nastase, Vivi ; Strube, Michael: Distinguishing between Instances and Classes in the Wikipedia Taxonomy. In: (Bechhofer u. a., 2008), S. 376–387. – ISBN 978-3-540-68233-2

[Zlatic u. a. 2006] Zlatic, V. ; Bozicevic, M. ; Stefancic, H. ; Domazet, M.: *Wikipedias: Collaborative web-based encyclopedias as complex networks.* Jul 2006. – URL `http://arxiv.org/abs/physics/0602149`

Eidesstattliche Erklärung

Ich versichere an Eides statt durch meine Unterschrift, dass ich die vorstehende Arbeit selbständig und ohne fremde Hilfe angefertigt und alle Stellen, die ich wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen habe, als solche kenntlich gemacht habe, mich auch keiner anderen als der angegebenen Literatur oder sonstiger Hilfsmittel bedient habe. Die Arbeit hat in dieser oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

Regensburg, den 15. Dezember 2009 .....
(Vorname Nachname)

A. Tabelle *term_cooccurrence_frequencies*

Spaltenname	Datentyp	Beschreibung
rev_id	bigint	Revisions-ID des Artikels, in dem die Kookkurrenz vorkommt
left_surface	varchar (255)	Oberflächen -Form des linken Terms
left_lemmata	varchar (255)	Grundformen des linken Terms
left_part_of_speech	smallint	Part-of-Speech-Tag des linken Terms
left_count_std_corpus	smallint	Okkurrenz-Häufigkeit des linken Terms (Ankertext)
left_count_link_corpus	smallint	Okkurrenz-Häufigkeit des linken Terms (Links)
right_surface	varchar (255)	Oberflächen-Form des rechten Terms
right_lemmata	varchar (255)	Grundformen des rechten Terms
right_part_of_speech	smallint	Part-of-Speech-Tag des rechten Terms
right_count_std_corpus	smallint	Okkurrenz-Häufigkeit des rechten Terms (Ankertext)
right_count_link_corpus	smallint	Okkurrenz-Häufigkeit des rechten Terms (Links)
coocc_count_std_offset1	smallint	Kookkurrenz-Häufigkeit (Abstand: *ein* Token; *mit* Funktionswörtern; Ankertext)
coocc_count_link_offset1	smallint	Kookkurrenz-Häufigkeit (Abstand: *ein* Token; *mit* Funktionswörtern; Links)
coocc_count_std_content_offset1	smallint	Kookkurrenz-Häufigkeit (Abstand: *ein* Token; *ohne* Funktionswörter; Ankertext)
coocc_count_link_content_offset1	smallint	Kookkurrenz-Häufigkeit (Abstand: *ein* Token; *ohne* Funktionswörter; Links)
⋮		
coocc_count_std_offset4	smallint	Kookkurrenz-Häufigkeit (Abstand: *vier* Token; *mit* Funktionswörtern; Ankertext)
coocc_count_link_offset4	smallint	Kookkurrenz-Häufigkeit (Abstand: *vier* Token; *mit* Funktionswörtern; Links)
coocc_count_std_content_offset4	smallint	Kookkurrenz-Häufigkeit (Abstand: *vier* Token; *ohne* Funktionswörter; Ankertext)
coocc_count_link_content_offset4	smallint	Kookkurrenz-Häufigkeit (Abstand: *vier* Token; *ohne* Funktionswörter; Links)
creation_timestamp	timestamp	Zeitstempel der Erstellung des Datensatzes

B. Literatur-Analyse wissenschaftlicher Arbeiten zur Wikipedia

Arbeit	Link-Graph	Link-Anchor	Kategorien	Artikeltext	Kontext	IW-Links	Revisionen	Infoboxen	Weiterleitungen	Disambiguierung
Auer u. a. (2007); Auer und Lehmann (2007)	–	–	•	–	–	–	–	•	–	–
De Smet und Moens (2009)	–	–	–	•	–	–	–	•	–	–
Gabrilovich und Markovitch (2007, 2009)	•	•	–	•	–	–	–	–	•	–
Ganjisaffar u. a. (2009)	–	–	–	–	–	–	•	–	–	–
Capocci u. a. (2006)	•	–	–	–	–	•	–	–	–	–
Athenikos und Lin (2009)	•	–	–	–	–	–	–	•	–	–
Gaoying Cui und Chen (2008)	–	–	•	–	–	–	–	–	–	–
Han und Zhao (2009)	•	•	•	–	–	–	–	–	•	–
Hu u. a. (2007)	–	–	–	•	–	–	•	–	–	–
Huang u. a. (2008, 2009)	•	•	–	–	•	•	–	–	–	–
Iftene und Balahur-Dobrescu (2008)	–	–	–	•	•	–	–	–	–	–
Kassner u. a. (2008)	–	–	•	•	–	–	–	–	–	–
Kinzler (2008)	•	•	•	•	–	•	–	–	•	•
Mehler (2006, 2008)	•	–	•	–	–	–	–	–	•	–
Mihalcea (2007)	–	•	–	•	•	–	–	–	–	–
Ruiz-Casado u. a. (2005a,b)	–	–	–	•	–	–	–	–	–	–
Schönhofen (2006)	–	–	•	–	–	–	–	–	•	–
Schönhofen (2008)	•	–	–	•	–	–	–	–	•	•
Ortega (2009)	–	–	–	–	–	–	•	–	–	–
Suh u. a. (2009)	–	–	–	–	–	–	•	–	–	–

Arbeit	Link-Graph	Link-Anchor	Kategorien	Artikeltext	Kontext	IW-Links	Revisionen	Infoboxen	Weiterleitungen	Disambiguierung
Syed u. a. (2008)	•	–	•	•	–	–	–	–	–	–
Tan u. a. (2009)	•	–	•	•	–	–	–	•	–	–
Wöhner und Peters (2009)	–	–	–	–	–	–	•	–	–	–
West u. a. (2009)	•	–	–	–	–	–	–	–	–	–
Zlatic u. a. (2006)	•	–	–	–	–	–	–	–	–	–
Zirn u. a. (2008)	•	–	•	•	–	–	–	–	–	–
Adafre und de Rijke (2006)	•	–	–	•	•	•	–	–	–	–
Nothman u. a. (2009)	–	•	•	•	–	–	–	–	•	•
Ramanathan u. a. (2009)	–	–	–	•	–	–	–	–	–	–
Roth und Schulte im Walde (2008)	•	–	–	•	–	–	–	–	–	–
Potthast u. a. (2008)	–	–	–	•	–	•	–	–	–	–
Milne und Witten (2008)	–	•	–	•	•	–	–	–	–	–
Milne (2007)	•	–	–	–	–	–	–	–	•	•
Buriol u. a. (2006)	•	–	–	–	–	–	•	–	–	–
Buscaldi und Rosso (2007)	–	–	–	•	–	–	–	–	–	–
Holloway u. a. (2007)	–	–	•	–	–	–	•	–	–	–
Ponzetto und Strube (2007); Ponzetto und Navigli (2009)	–	–	•	•	–	–	–	–	–	–
Wu und Weld (2007)	•	–	–	•	–	–	–	–	–	–
Kittur u. a. (2008)	–	–	–	–	–	–	•	–	–	–
Ito u. a. (2008)	–	–	–	–	•	–	–	–	–	–