Entitätenextraktion und Relevanzbewertung mit Hilfe von semantischen Datenquellen und APIs.

Stephan Hamberger

MASTERARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

Interactive Media

in Hagenberg

im Oktober 2012

Erklärung

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

Hagenberg, am 8. Oktober 2012

Stephan Hamberger

Inhaltsverzeichnis

Vorwort

Diese Masterarbeit entstand im Rahmen des Forschungsprojekts „SemantLink“ – Linked Data im Unternehmenseinsatz, einem disziplinübergreifenden Projekt der Studiengänge Interactive Media und Kommunikation, Wissen, Medien am Campus Hagenberg der FH Oberösterreich in Zusammenarbeit mit der FH OÖ Forschungs- und Entwicklungs GmbH. In diesem Projekt ging es um die Strukturierung und Klassifikation von Inhalten für die Wissensverwaltung in Unternehmen, insbesondere um die Optimierung von Strukturierungsprozessen um passende Navigationsstrukturen und Suchzugänge zur Überwindung der Informationsflut aus Internetanwendungen, Wikis, Blogs und dergleichen zu schaffen. Dabei sollten nicht nur herkömmliche Verfahren zur automatischen Dokumentenklassifikation und Suche eingesetzt, sondern auch eine breitere Wissensbasis der verwalteten Dokumente geschaffen werden. Dazu sollten Dokumente (teil)automatisiert mit Metadaten und Zusatzinformationen angereichert werden.

„Linked Data“, eine weltweit verteilte Wissensbasis aus dem semantischen Web, das in Form eines standardisierten Datenformats eine globale Daten-Infrastruktur anbietet, soll diesem Anreicherungsprozess als Datenpool dienen. Daraus sollen neue integrierte Sichtweisen auf zunächst zusammenhangslos verteilte Daten gefunden werden, was Recherche-Zeiten und Fehlerpotential minimiert sowie neue Möglichkeiten für Business Intelligence, Knowledge Management oder CRM bietet.

Als besonderes technisches Kernstück dieser Herangehensweise dient eine ausgereifte Computerlinguistik-Applikation. Um im Wissensmanagement benötigte Daten automatisiert zu erfassen ist eine Textanalyse unumgänglich. Um eine für dieses Projekt optimierte Basis zu schaffen, wurden deshalb neue Methoden und Ansätze in der Computerlinguistik erforscht. Als Grundgedanke stand die Verwendung von externen Datenquellen im Vordergrund. Dieser Aspekt liegt der vorliegenden Arbeit zugrunde. Dabei wird untersucht, ob und wie externe Datenquellen zur Performanzsteigerung von Applikationen in der Computerlinguistik herangezogen werden können. Im Speziellen wird dabei die Verwendung von externen Datenquellen in der Entitätenextraktion und Relevanzbewertung betrachtet. Darüberhinaus wurde versucht auch die in diesem Bereich auftretenden Limitationen aufzuzeigen. Selbstverständlich gibt es darüberhinaus eine Vielzahl weiterer potentieller

Verwendungsmöglichkeiten in diesem Bereich, die in dieser Arbeit nicht explizit behandelt werden konnten.

Ein besonderer Dank gilt der FH OÖ Forschungs- und Entwicklungs GmbH und der FH Oberösterreich, Campus Hagenberg im Speziellen Andreas Stöckl, Thomas Kern sowie Doris Divotkey für das Ermöglichen dieses Projekts. Bedanken möchte ich mich auch bei Kasra Seirafi für die Zusammenarbeit im SemantLink Projekt. Ein weiterer Dank gilt darüberhinaus Andreas Stöckl für die Betreuung dieser Masterarbeit sowie den Professoren des Studiengangs Interactive Media. Nicht zuletzt möchte ich mich auch bei meiner Familie für die moralische Unterstützung und das Korrekturlesen bedanken.

Kurzfassung

Die Verwendung in der Computerlinguistik von Daten aus dem *Semantic-* und *Social-Web*, welche eine wachsende Wissenbasis darstellen, die zunehmend über APIs oder semantische Technologien zugänglich und maschinell lesbar gemacht wird, ist Gegenstand dieser Arbeit.

Zum einen wurde versucht einen Ansatz zu finden, in welchem externe Datenquellen zur Verbesserung der Ergebnisse von bestehenden Anwendungen herangezogen werden können. Dabei hat sich der Einsatz von der semantischen Wissensbasis DBpedia bei der *Entitätenextraktion* als am sinnvollsten herausgestellt. Zudem bildet die *Entitätenextraktion* den Grundstein der *Informationsextraktion*. Somit wird möglichst weit unten in der *Computerlinguistik* Pipeline angesetzt, um eine optimierte Basis für eine darauf aufbauende semantische Analyse zu schaffen. Es hat sich herausgestellt, dass eine Kombination von bestehenden Ansätzen und der Verwendung von externen Datenquellen das beste Ergebnis liefert.

Zum anderen ermöglicht die Verwendung von externen Datenquellen die Erstellung von völlig neuen Applikationen. Eine in dieser Arbeit verfolgte Methode stellt die Relevanzbewertung von Enitäten dar. Basierend auf Informationen von DBpedia, Facebook Open Graph und der Google AdWords API wird sowohl eine Globalrelevanz als auch eine Kontextrelevanz berechnet.

Die Implementierung dieser Ansätze im Rahmen der SemantLink Applikation hat gezeigt, dass sowohl eine Performanzsteigerung der Entitätenextraktion als auch eine sinnvolle Relevanzbewertung möglich ist. Das World Wide Web kann somit als computerlinguistische Ressource nicht nur als Korpus zur Analyse, sondern auch durch die daraus resultierende kollektive Intelligenz zur Verbesserung der Methoden der Computerlinguistik selbst herangezogen werden.

Abstract

Improving natural language processing tasks by the use of data from *Semantic-* and *Social Web*, which represents a growing knowledge base made accessible through APIs or semantic technologies, is the core of this master's thesis.

For one thing external data was used to enhance the results of existing applications and tasks. The use of the semantic database DBpedia has proved to be most effective in named-entity recognition. Moreover the named-entity recognition represents the basis of the information extraction. By starting as low down as possible in the natural language processing pipeline an optimized basis for the further semantic analysis is created. A combination of existing approaches and the use of external data sources have proved to produce the best results.

For another the use of external data sources enables the production of completely new applications. Relevance rating of named entities is one of the methods used in this thesis. Based on information from DBpedia, Facebook Open Graph and Google AdWords API, both a global and a contextual relevance are calculated.

The implementation of these methods in the course of the SemantLink application has shown that both an increase in the performance of the named-entity recognition and a meaningful relevance rating are possible. The World Wide Web as a natural language processing resource can thus not only be used as a corpus for analysis but also by the resulting collective intelligence for the improvement of methods of natural language processing tasks.

Kapitel 1

Einleitung

1.1 Motivation

In der heutigen Informationsgesellschaft, nimmt der Einfluss der Computerlinguistik auf unser tägliches Leben ständig zu. Beim Surfen im Internet, bei der Verwendung eines Smartphones oder eines Computers kommt man fast unvermeidlich mit Applikationen aus dieser Wissenschaft in Berührung. Beispiele für verbreitete Applikationen sind *Rechtschreibkorrektur*, *Grammatiküberprüfung*, *statistische Informationen*, *Lexikographie*, *Übersetzung* und *Sprachsteuerung*. Dabei handelt es sich allerdings nur um die offensichtlichsten Anwendungsszenarien, denn vielen anderen alltäglichen Applikationen wie der Suche und den Support-Systemen liegen ebenfalls oft computerlinguistische Algorithmen zugrunde.

Die Verbesserung der bestehenden Algorithmen hat somit nicht nur einen bedeutenden wissenschaftlichen sondern auch praktischen Wert. Mit der zunehmenden Technologisierung unseres Alltags nimmt auch die Bedeutung der Computerlinguistik zu. Der Begriff *Ubiquitous Computing* beschreibt, dass sich der Computer als Gerät immer mehr in den Hintergrund verschiebt und schließlich unsichtbar wird und sich nahtlos in seine Umgebung einfügt. Schon heute nimmt die Bedeutung von immer kleiner werdenden Endgeräten deutlich zu. Die immer kleiner werdende Bedienfläche und Integration in die Umgebung erfordern neue Möglichkeiten zur Eingabe und Kommunikation mit dem Gerät. Eine der verbreitetsten Alternativen zur Eingabe über ein Keyboard ist die Spracheingabe. Während einfache Kommandos wie die Aufforderung zum Tätigen eines Anrufs schon seit Jahren in Mobiltelefonen integriert sind, sind dank verbesserter Algorithmen in den Bereichen Computerlinguistik, künstliche Intelligenz, maschinelles Lernen und Wissensrepräsentation mittlerweile komplexere Anweisungen bei der Kommunikation mit einem mobilen Endgerät möglich. So hat zum Beispiel das Unternehmen *Apple Inc.* am 28. April 2010 das Unternehmen *Siri Inc.* und sein gleichnamiges Produkt *Siri* erworben, eine iPhone Applikation, welche die Tätigkeit

eines persönlichen Assistenten erfüllt.[1] Seit dem *iPhone 4S*, welches am 4. Oktober 2011 vorgestellt wurde, ist *Siri* ein fester Bestandteil von Apples mobilem Betriebssystem *iOS*.[2] Neben der Verwendung von integralen Funktionen des Betriebssystems wie zum Beispiel die Erstellung einer Erinnerung, dem Senden einer Nachricht oder das Auffinden eines Kontakts, lassen sich über die Integration von diversen Drittanbietern wie zum Beispiel dem online Tischreservierungsdienst *Open Table*, der Online-Bewertungs Plattform *Yelp*, der Wissensmaschine *Wolfram Alpha* sowie der Suchmaschinen *Google* und *Bing* diverse Informationen aus dem *World Wide Web* abrufen.

Dies wird durch eine zu Grunde liegende, immer größer werdende Wissensbasis ermöglicht. Seit der Kommerzialisierung des Internets in den 1990er Jahren kam es zu einer rasanten Weiterentwicklung dieses Mediums. Neben der Verbesserung von Browser Engines, der Einführung von zahlreichen neuen Technologien, der Erweiterung des Adressraums und der Veränderung und Mobilisierung von Endgeräten hat vor allem die Erschaffung von Plattformen, Tools und Webservices zur zunehmend vereinfachten Generierung von Inhalten im Web geführt. Die Zugänglichkeit zu diesen Daten mit Hilfe von APIs und semantischen Technologien ermöglicht nun auch eine automatisierte Informationsextraktion, welche zur Wissensgenerierung herangezogen werden kann. Die steigende Quantität an verfügbaren Inhalten hat aber nicht nur Vorteile. Sie führt zu einer Informationsüberflutung und zu Inkonsistenz und somit zu einer zunehmend erschwerten Überprüfbarkeit des Wahrheitsgehalts. Deshalb werden in Zukunft Filter- und Suchmechanismen, Bewertungsmöglichkeiten sowie die Aufbereitung von Wissen noch eine viel größere Rolle einnehmen als dies heute schon der Fall ist. Diesbezüglich gewinnt vor allem die *Computerlinguistik* und insbesondere die *Informationsextraktion* an Bedeutung, da diese sowohl die semantische Analyse von natürlicher Sprache als auch die Extraktion von Wissen aus unstrukturierten Inhalten ermöglichen. Gleichzeitig sollen die Inhalte im Web optimalerweise nicht nur als Korpus zur Analyse sondern auch zur Verbesserung der Methoden der Computerlinguistik selbst herangezogen werden. Im Idealfall könnte somit ein selbstlernendes System entstehen, welches durch die zunehmende Verfügbarkeit von Inhalten immer bessere Ergebnisse liefert.

Da die *Entitätenextraktion* den Grundstein für die *Informationsextraktion* darstellt, beschäftigt sich diese Arbeit mit dem Einsatz von externen Datenquellen in diesem Gebiet. Somit wird möglichst weit unten in der typischen *Computerlinguistik* Pipeline angesetzt, um eine optimierte Basis für eine weiter darauf aufbauende semantische Analyse zu schaffen. Ein weiterer fundamentaler Mechanismus, welcher Filter- und Suchmechanismen verbessern soll, ist die Relevanzbewertung. Auch hier ist es interessant für diese Be-

[1] http://scobleizer.com/2010/04/28/breaking-news-siri-bought-by-apple/

[2] http://www.telegraph.co.uk/technology/apple/8804922/Apple-iPhone-event-live.html

wertung auf Daten aus dem *World Wide Web* zurückzugreifen. Hierfür sind vor allem Daten aus dem *Social Web* und von Suchmaschinen interessant. Diese Ansätze zur Verwendung der von Benutzern generierten Inhalten und die daraus resultierende kollektive Intelligenz zur *Entitätenextraktion* und Relevanzbewertung stellt somit eine besondere Herausforderung dar.

1.2 Zielsetzung

In den Vorarbeiten zu dieser Arbeit wurden verschiedenste Ansätze zur Verwendung der von benutzergenerierten Inhalte in der *Computerlinguistik* untersucht. Dabei wurden sowohl grundlegende computerlinguistische Verfahren wie Part-of-speech Tagging, Stoppwortsuche, Morphologische Analyse, Entitätenextraktion als auch spezifische Anwendungen wie Sentimentanalyse, Kategorisierung, automatisiertes Tagging, Aktienanalyse, Zukunftsvoraussagen, automatisierte Erstellung von Zusammenfassungen sowie Konzeptextraktion in die Evaluierung miteinbezogen. Gleichzeitig wurde nach relevanten Datenquellen gesucht, welche per API oder SPARQL Endpoint zugänglich sind. Einige Beispiele dafür sind Wolfram Alpha[3], Yelp[4], eHow[5], Yahoo! Answers[6], Goole Places[7], Facebook[8], Twitter[9], Google+[10], Linked Life Data[11], DBpedia[12], Freebase[13], Google AdWords[14] und Google Translate[15].

Bei der Themenfindung wurde darauf geachtet einen Teilaspekt zu wählen, welcher im Rahmen meines *Interactive Media* Studiums an der *FH Oberösterreich* am *Campus Hagenberg* sowohl als Masterprojekt mit Praxisbezug im Rahmen des SemantLink Projekts als auch als Masterarbeit umgesetzt werden konnte. Gleichermaßen wurde versucht ein Thema zu wählen, bei dem ein Aspekt betrachtet werden kann, welcher bisher in wissenschaftlichen Publikationen nur marginal oder noch gar nicht untersucht wurde. Zu dem sollte ein möglichst weitreichendes Gebiet gewählt werden, welches optimalerweise auch andere computerlinguistische Anwendungen verbessern kann.

Letztendlich haben sich die *Entitätenextraktion* und die darauf aufbauende *Relevanzbewertung* für diesen Zweck als optimales Forschungsumfeld für

[3] http://www.wolframalpha.com/
[4] http://www.yelp.com/
[5] http://www.ehow.com/
[6] http://answers.yahoo.com/
[7] http://www.google.com/places/
[8] https://www.facebook.com/
[9] https://www.twitter.com/
[10] https://plus.google.com/
[11] http://www.linkedlifedata.com/
[12] http://www.dbpedia.org/
[13] http://www.freebase.com/
[14] http://adwords.google.com/
[15] http://translate.google.com

den Einsatz von externen Datenquellen in der *Computerlingustik* ergeben. Die *Entitätenextraktion* stellt den Grundstein für die *Informationsextraktion* dar. Viele weitere Algorithmen wie zum Beispiel das Auffinden von Relationen, der Aufbau von Wissensbasen, *Textklassifikation*, *Suche* oder *Textclustering* basieren auf den Ergebnissen der *Entitätenextraktion*. Als externe Datenquellen können unter anderem diverse semantische Datenquellen wie *DBpedia* oder *Freebase* herangezogen werden.

Darüberhinaus könnten sich externe Datenquellen sehr gut zur Bewertung der einzelnen Entitäten verwenden lassen. Hierfür sind vor allem die Daten aus dem *Social Web* interessant: Die Anzahl der „Gefällt mir“ und „Sprechen darüber“ einer bestimmten Facebook Page, die Anzahl der Erwähungen auf Twitter oder die Follower eines Twitter Profiles könnten als Messwert zur Relevanzbewertung dienen. Neben den Daten aus dem *Social Web* könnten des Weiteren die Suchhäufigkeit eines Begriffs oder der durchschnittliche Gebotspreis bei *Google AdWords* zur Bewertung der Relevanz herangezogen werden. Daraus ergibt sich die folgende Forschungsfrage, welche im Rahmen dieser Masterarbeit behandelt werden soll: „Wie können semantische Datenquellen (DBpedia) und APIs (Google AdWords und Facebook) zur Verbesserung der Entitätenextraktion und Relevanzbewertung in der Computerlinguistik herangezogen werden?“

1.3 Inhaltlicher Aufbau

Der inhaltliche Aufbau dieser Arbeit gliedert sich in 4 Hauptteile. Nach der Einleitung wird im Kapitel 2 eine kurze Einführung in die *Computerlinguistik und Texttechnologie* gegeben. Neben den Aspekten der Computerlinguistik soll im Kapitel Grundlagen (2.1.1) ein Einblick in die formalen Grundlagen gegeben werden. Klassische computerlinguistische Methoden (2.1.2) wie *Morphologie* oder *Syntax und Parsing* werden ebenfalls erläutert wie typische Ressourcen (2.1.3). Im Einführungskapitel (2) soll des Weiteren Einblick in die Disziplin der *Informationsextraktion* (2.2) gegeben werden. Hier werden die Grundlagen von *Mustererkennung* (2.2.2), *Entitätenextraktion* (2.2.3) sowie Koreferenz und Relationen (2.2.4) erläutert.

Das Kapitel 3 beschäftigt sich damit, wie Inhalte aus dem *World Wide Web* als computerlinguistische Ressource verwendet werden können und bildet somit den theoretischen Hauptteil dieser Arbeit. Zu Beginn wird im Abschnitt Related Work (3.1) ein Überblick über bestehende wissenschaftliche Arbeiten im verwandten Themenumfeld gegeben. Nachfolgend werden die in der Arbeit betrachteten Datenquellen (3.2) vorgestellt und ihre Verwendungsmöglichkeiten evaluiert. Dabei werden sowohl semantische Datenquellen (3.2.1) wie *DBpedia* und Datenquellen, welche per API zugänglich sind, verwendet. Einen wichtigen Bestandteil bilden außerdem die Limitationen (3.2.4), welche externe Datenquellen mit sich bringen. Nachfolgend

werden die beiden Methoden zur *Entitätenextraktion* (3.3) und zur *Relevanzbewertung* (3.4) näher vorgestellt.

Den praktischen Hauptteil dieser Arbeit bildet das Kapitel 4. Hier wird der im Rahmen dieser Masterarbeit entwickelte Prototyp in *GATE* (General Architecture for Text Engineering) vorgestellt und auf spezifische Implementierungsdetails eingegangen. Abschließend wird im Kapitel 5 eine Evaluierung basierend auf dem *F1 Score* durchgeführt und die Ergebnisse der Prototypenimplementierung mit herkömmlichen Methoden und anderen Algorithmen verglichen. In den Schlussbemerkungen (Kapitel 6) wird ein abschließendes Fazit gezogen und ein Ausblick auf weitere Forschungsmöglichkeiten gegeben.

Kapitel 2

Grundlagen

2.1 Computerlinguistik und Texttechnologie

Die *Computerlinguistik* beschäftigt sich mich der maschinellen Verarbeitung von natürlicher Sprache in Form von Text- oder Sprachdaten. Klassifizieren lässt sie sich als Teilbereich der künstlichen Intelligenz und als Schnittstellendisziplin zwischen *Linguistik* und *Informatik* [4, S. 1].

In der heutigen Informationsgesellschaft, nimmt der Einfluss der Computerlinguistik auf unser tägliches Leben ständig zu. Ob beim Surfen im Internet, bei der Verwendung eines Smartphones oder eines Computers – man kommt fast unvermeidlich mit Applikationen aus dieser Wissenschaft in Berührung. Beispiele für verbreitete Applikationen sind *Rechtschreibkorrektur*, *Grammatiküberprüfung*, *Statistische Informationen*, *Lexikographie*, *Übersetzung* und *Sprachsteuerung*. Dabei handelt es sich allerdings nur um die offensichtlichsten Anwendungsszenarien, denn vielen anderen alltäglichen Applikationen wie Suche und Support-Systemen liegen ebenfalls oft computerlinguistische Algorithmen zugrunde.

Die *Informationsextraktion* versteht sich als Anwendung von kombinierten Verfahren aus der künstlichen Intelligenz und der *Computerlinguistik*. Sie versucht strukturierte Informationen aus natürlicher Sprache zu gewinnen um diese brauchbarer für Informationsverarbeitungsprozesse zu machen [15]. In diesem Kapitel werden für die Computerlinguistik benötigte *formale Grundlagen* (2.1.1) wie *Mengenlehre*, *Aussagen-*, *Prädikaten-* und *Typenlogik*, *Grammatiken*, *Graphentheorie*, *Statistische Grundlagen* sowie *Texttechnologie* erläutert. Auf eine genaue Einführung der jeweiligen Grundlagen wird verzichtet, da deren Kenntniss vorausgesetzt wird. Vielmehr sollen die in dieser Arbeit benötigten Grundlagen lediglich aufgelistet werden.

Anschließend wird ein kurzer Einblick in die typischen Methoden (2.1.2) und Ressourcen (2.1.3) gegeben. Auch hier soll nicht die Anforderung der Vollständigkeit erfüllt werden, sondern eine Einführung in die benötigten Grundkenntnisse für den Hauptteil dieser Arbeit gegeben werden.

2.1.1 Grundlagen

Computerlinguistische Methoden basieren auf speziellen Grundlagen der Mathematik, der Informatik und der Linguistik. In diesem Kapitel werden deshalb einige dieser Grundlagen aufgelistet. Auf eine genauere Erklärung wird verzichtet, da deren Kenntnis vorausgesetzt wird.

Mathematische Grundlagen

Für ein formales Vorgehen im Rahmen der Computerlinguistik sind Grundlagen der *Mengenlehre* und der *Logik* unverzichtbar. Gerade die mengentheoretischen Konzepte finden nahezu in allen Bereichen der Computerlinguistik Verwendung. Dies ist zum Beispiel bei der Bestimmung von Wahrscheinlichkeiten für Mengen von Ergebnissen in der Statistik oder bei der Definition einer formalen Sprache als Menge von Zeichenketten der Fall [4, S. 28]. Mehr dazu findet sich in in [4, Kap. 2]. Die Logik als Lehre des Schlussfolgerns wird in der Computerlinguistik meist bei der Anwendung von statischen Regeln und Grammatiken verwendet. Spezielle Bedeutung kommt hier der Aussagen-, Typen- und Prädikatenlogik zu. Mehr dazu findet sich in [4, Kap. 2].

Automatentheorie und formale Sprache

Die *Automatentheorie* und die *formale Sprache* sind Teilgebiete der theoretischen Informatik, welche für die Computerlinguistik von großer Bedeutung sind [4, S. 66]:

> Für die maschinelle Verarbeitung natürlicher Sprache ist es notwendig die jeweilige Sprache bzw. die relevanten Ausschnitte in eine Hierarchie formaler Sprachen einzuordnen, um zu wissen, mit welchen Mitteln und wie effizient diese Sprache bzw. dieser Sprachausschnitt analysiert werden kann.

Die Automatentheorie befasst sich somit mit formalen Sprachen und Grammatiken. Während *Grammatiken* valide Sprachen erzeugen, werden Worte einer Sprache mittels *Automaten* erkannt. Dabei wird ein gegebenes Wort analysiert und anschließend entschieden, ob es zu einer von einer Grammatik festgelegten Sprache gehört oder nicht. Im einfachsten Fall wird bei der Erkennung nur mitgeteilt, ob das jeweilige Wort als Element der jeweiligen formalen Sprache akzeptiert wurde oder nicht. Neben der bloßen Information über Akzeptanz kann bei der Erkennung auch eine Ausgabe mit Zusatzinformationen erfolgen. Im diesem Fall werden die Automaten als *Maschinen* bezeichnet. [4, S. 70]. Mehr dazu findet sich in [4, Kap. 2].

Bei einer *formalen Sprache* steht im Gegensatz zu einer *konkreten Sprache* nicht die Kommunikation im Vordergrund, sondern die mathematische Verwendung. Ein typisches Beispiel für eine formale Sprache ist eine Program-

miersprache. Formale Sprachen bestehen aus einer bestimmten Menge von Zeichenketten, die aus einem Zeichenvorrat zusammengesetzt werden können und eignen sich zur mathematischen präzisen Beschreibung im Umgang mit Zeichenketten. So können zum Beispiel Dateneingaben, Zeichenketten oder eben ganze Programmiersprachen spezifiziert werden. Mehr dazu findet sich in [4, Kap. 2].

Grammatiken

Wenn Wörter einer bestimmten Sprache zu Sätzen zusammengefügt werden, gelten bestimmte Regeln. Alle Menschen kennen diese Regeln von frühester Kindheit an, meist ohne sie für ihre Muttersprache bewusst bestimmen zu können. Diese Regeln werden Grammatiken genannt und sind von Sprache zu Sprache größtenteils verschieden. Die Herausforderung der Computerlinguistik ist es, dieses Regelwissen so zu beschreiben, dass automatisierbare Verfahren der Analyse und der Erzeugung von Sätzen beschrieben werden können [10, S. 23]. Grammatiken beschreiben mit Hilfe von Relationen und Regeln an welcher Position und in welcher Reihenfolge verschiedene Wortarten auftreten dürfen. Dabei kann ein Wort in einem Satz zwei verschiedene Arten von Bezügen eingehen: *syntagmtische* und *paradigmatische*.

Definition Syntagmatische Relation [10, S. 23]: Bezüge, die ein Wort zu anderen vor oder nach ihm erscheinenden Wörtern in einer Wortgruppe aufweist, werden als *syntagmatische Relationen* bezeichnet. Die fundamentalste syntagmatische Relation ist die der unmittelbaren Nachbarschaft zweier Wörter. Eine Wortgruppe, deren Einheiten durch syntagmatische Relationen miteinander verbunden sind, nennt man *Syntagma*.

Definition Paradigmatische Relation [4, S. 24]: Bezüge, die ein Wort, das in einer Wortgruppe erscheint, zu anderen Wörtern aufweist, die an der gleichen Stelle in der Wortgruppe erscheinen könnten, werden als *paradigmatische Relationen* bezeichnet. Derartige Wörter können zu Gruppen zusammengefasst werden, die auch *Paradigma* genannt werden. Bei der Bildung von Gruppen können formale oder inhaltliche Kriterien geltend gemacht werden.

Grammatiken erzeugen valide Wörter und Sätze, die Menge der von einer Grammatik erzeugten Wörter bilden eine Sprache. Einer der wichtigsten Grundsätze ist dabei die Typisierung und Gruppierung von Wörtern in bestimmte Wortarten. Eine Auflistung der Wortarten im Deutschen findet sich in Tabelle 2.1.

Bei den Definitionen der Wortarten kann man vor allem zwischen veränderlichen wie Nomen, Verb, Adjektiv, usw. und unveränderlichen Wortarten, wie Adverb, Präposition und Partikel unterscheiden. Des Weiteren gibt es noch „offene“ Wortartengruppen also Gruppen, in welchen bei Bedarf neue Wörter gebildet werden können (z.B.:Nomen, Verb und Adjektiv) als auch

Tabelle 2.1: Definition: Wortarten im Deutschen [10, S. 24].

Verb (V): Verben sind diejenigen Wörter, die eine finite Form aufweisen können. Eine finite Verbform wird gebildet durch die erste, zweite oder dritte Person im Singular oder Plural. Beispiel: *lache, lachst, lacht.* Neben einer finiten Verbform kann ein Verb auch infinite Formen annehmen: *lachend* (Partizip I), *gelacht* (Partizip II), *lachen* (Infiitiv).
Nomen (N): Nomina sind diejenigen Wörter, die sich nach Kasus (Nominativ, Genitiv, Dativ, Akkusativ) und normalerweise nach Numerus (Singular, Plural) verändern (deklinieren) lassen und dabei ein unveränderliches Genus (Maskulinum, Femininum, Neutrum) besitzen. Beispiele: *Buch, Buches, Bücher.*
Determination (D): Determination sind Wörter, die sich hinsichtlich Kasus, Numerus und Genus verändern lassen, die vor einem Nomen erscheinen und mit diesem zusammen eine Nominalphrase (Wortgruppe mit Nomen als Kopf) bilden: *das* Buch, *ein* Buch, *drei* Bücher, *kein* Buch. Dabei passt sich das Determination in seiner Flexion der des Nomens an.
Pronomen (PRO): Pronomina sind diejenigen Wörter, die sich hinsichtlich Kasus, Numerus, Genus und Person deklinieren lassen und im Satz anstelle einer Nominalphrase erscheinen können: *dieses* Buch → *es/dieses/jenes*, Peter lachte → *er/ich/niemand lachte.*
Adjektiv (A): Adjektive lassen sich nach Kasus, Numerus und Genus deklinieren und können zwischen einem Artikel und einem Nomen stehen: das *alte* Buch, die *alten* Bücher. Viele Adjektive können kompariert (gesteigert) werden: *alt*, *älter* (Komparativ), (am) *ältesten* (Superlativ). Eine Besonderheit deutscher Adjektive besteht darin, dass sie sich auch in Abhängigkeit von der Art des Artikels verändern: das *alte* Buch (Defizit), ein *altes* Buch (indefinit).
Adverb (ADV): Adverbien sind nicht veränderliche Wörter, die am Satzanfang stehen und als Antwort auf Sachfragen dienen können: *Wo* ist das Buch? *Hier/dort/oben/daneben.*
Präposition (P): Präpositionen sind unveränderliche Wörter, die sich mit einem Nomen in einem bestimmten Kasus verbinden, der von ihnen festgelegt wird: *wegen* des Buches, *neben* dem Buch, *auf* das Buch.
Partikel (PTL): Partikel sind alle anderen unveränderlichen Wörter. In dieser großen und heterogenen Gruppe können weitere Untergruppen gebildet werden, etwa die der Modalpartikeln (*sicherlich, vielleicht*), der Negationpartikeln (*nicht, gar nicht*), der Subjunktoren (SUB; *weil, wenn, das nachdem*) und der Konjunktoren (KON; *und, oder, sowohl ... als auch*).

geschlossene Wortarten Gruppen, bei denen der Bestand im gegenwärtigen Deutsch feststeht (z.B. Präpositionen Konjunktoren).

Im Rahmen der Computerlinguistik wird dabei eine Menge R an Regeln definiert, mit deren Hilfe eine Grammatik validiert werden kann. Grammatiken arbeiten dabei binär, denn entweder ist eine bestimmte Zeichenkette generierbar und gehört damit zu der von der Grammatik erstellen Sprache oder nicht [4, S. 67].

Formal werden Grammatiken als Quadrupel definiert. Sie bestehen aus zwei Alphabeten, einem Alphabet $\sum$ so genannter *Terminalsymbole* und einem Alphabet ϕ von *Nichtterminalsymbolen* oder *Variablen*. Beide Mengen sind disjunkt. Des Weiteren wird ein Startsymbol $S \in \phi$ benötigt sowie eine Regelmenge R zur Generierung der aus Terminalsymbolen bestehenden Zeichenketten [4, S. 67]:

Eine Grammatik $G = \langle \phi, \sum, R, S \rangle$ besteht somit aus:

1. Einem Alphabet ϕ von Nichtterminalsymbolen,
2. Einem Alphabet $\sum$ von Terminalsymbolen mit $\phi \bigcap \sum = 0$,
3. Einer Menge $R \subseteq \Gamma^* \times \Gamma^*$ von Ersetzungsregeln $\langle \alpha, \beta \rangle$ (Γ ist das Gesamtalphabet $\phi \bigcup \sum$, wobei zusätzlich gilt : $\alpha \neq \epsilon$ und $\alpha \neq \sum^*$,
4. Einem Startsymbol $S \in \phi$

Diese Definition einer Grammatik legt eine so genannte allgemeine Regelgrammatik (auch Typ-0-Grammatik genannt) fest, deren einzige Bedingung für die einzelnen Regeln in der Regelmenge ist, dass mindestens ein nichtterminales Symbol durch eine beliebige Zeichenkette über dem Gesamtalphabet Γ ersetzt wird. Die von einer Grammatik beschriebene formale Sprache wird als die Menge derjeniger Zeichenketten festgelegt, die durch Regelanwendungen aus dem Startsymbol abgeleitet werden können [4, S. 68]:

Sei $G = \langle \phi, \sum, R, S \rangle$ eine Grammatik und seien $u, v \in (\phi \bigcup \sum)^* = \Gamma^*$.

1. v ist aus u *direkt ableitbar*, falls gilt: $u = u_1 w u_2, v = u_1 z u_2 und w \rightarrow z$ ist eine Regel aus R.
2. v ist aus u *ableitbar*, falls es Wörter $u_0, ..., u_k$ gibt ($k \leq 0$), so dass $u = u_0, v = u_k$ und $u_{i-1} \rightarrow u_i (1 \leq i \leq k)$ gilt. v ist also aus u ableitbar, wenn es Zwischenwörter gibt, die jeweils direkt ableitbar sind.

Ableitungen lassen sich graphisch auch als besondere Graphen in Form von Bäumen darstellen.

Graphentheorie

Die Graphentheorie stellt eine unentbehrliche Grundlage zur Beschreibung linguistischer Objekte und Strukturen dar. Eine Besonderheit dabei ist die Möglichkeit hierarchische Beziehungen zu beschreiben. Ein Beispiel im Rahmen der formalen Sprache und Grammatik ist ein Ableitungsbaum. Graphen spielen aber auch in der *Morphologie* und insbesondere der *Syntax* eine ent-

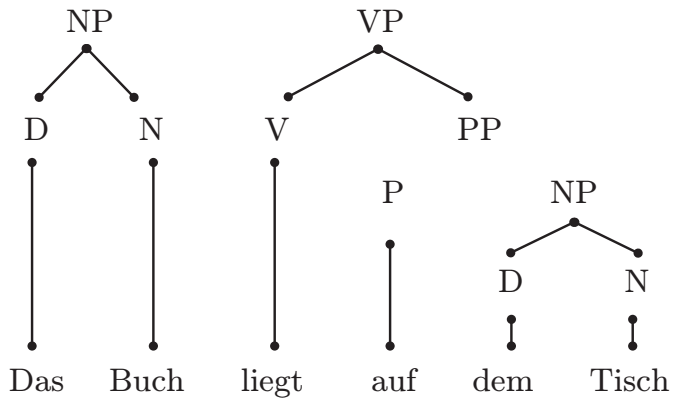

Abbildung 2.1: Beispiel für eine Ableitungsbaum.

scheidende Rolle [4, S. 94]. Mehr zur Graphentheorie findet sich in [4, Kap. 2].

Statistische Grundlagen

Statistische Grundlagen und vor allem die Wahrscheinlichkeitstheorie spielen u.a. eine bedeutende Rolle bei der Entwicklung und Verwendung von probabilistischen Grammatiken, bei der automatischen Annotierung von Texten mit Wortklassen und anderen linguistischen Merkmalen auf Wortebene, dem sogenannten *Tagging* [4, S. 114]: Zu den erfolgreichsten wahrscheinlichkeitsbasierten Ansätzen in der Computerlinguistik gehören *Hidden-Markov-Modelle*, die insbesondere für Tagging Aufgaben eingesetzt werden. Auch wenn diese seit Ende der 90er Jahre durch komplexere statistische Methoden abgelöst wurden, werden sie aufgrund ihrer Effizienz und einfachen Implementierung auch heute noch in vielen Anwendungsgebieten eingesetzt. Mehr dazu findet sich in [4, Kap. 2].

Texttechnologie

Die Texttechnologie beschäftigt sich mit der linguistisch motivierten Informationsanreicherung und Verarbeitung digital verfügbarer Texte mittels standardisierter Auszeichnugsprachen [4, S. 159]: In allen texttechnologischen Anwendungen wird die Metasprache XML (*Extensible Markup Language*) eingesetzt, die die Definition beliebiger Auszeichnungssprachen erlaubt. XML ist also eine formale Sprache zur Spezifizierung konkreter Markup-Sprachen, die wiederum zur Auszeichnung (engl. Annotation) arbiträrer Informationseinheiten in textuellen Daten eingesetzt werden können. In computerlinguistischen Anwendungen geht es hierbei u.a. um die Auszeichnung (Annotation) linguistischer Informationen in Texten, die Verwendung von

XML als Datenaustauschformat zur einheitlichen Repräsentation von Ressourcen. Mehr dazu findet sich in [4, Kap. 2].

2.1.2 Methoden

Phonetik und Phonologie

Die Lautlehre wird üblicherweise in zwei Hauptbereiche gegliedert [4, S. 170-172]: Phonetik und Phonologie. Die Phonetik beschäftigt sich dabei mit allen Details der Lautproduktion, der Akustik der Lautübertragung und der Lautrezeption. Die Phonetik ist auf wortunterscheidende Lauteigenschaften, die Lautstrukturen und die Relationen zwischen den Lauten fokussiert. Die Computerphonetik und die Computerphonologie beschäftigen sich mit den lautsprachlichen Eigenschaften, Formen und Strukturen der ca. 7000 Sprachen weltweit. Besonderes Augenmerk gilt auch den unterschiedlichen Aussprachevarianten der einzelnen Sprachen, beispielsweise Hochdeutsch, österreichisches Deutsch, Schweizerdeutsch sowie regionale Dialekte. Die Vielfalt an Varianten der Phonologie und der Phonetik lässt sich mit relativ einfachen formalen Mitteln modellieren und operationalisieren: In erster Linie mit endlichen *Automaten* und *Transduktoren*, aber auch mit *Hiden-Markov-Moddellen (HMM)*. Sowohl die Phonetik und Phonologie als auch die Verarbeitung gesprochener Sprache werden in dieser Arbeit nicht benötigt, der Vollständigkeit halber werden diese allerdings angeführt. Mehr dazu findet sich in [4, Kap. 3].

Verarbeitung gesprochener Srache

Phonetische Modelle bilden die Grundlage für die Verarbeitung von gesprochener Sprache in der Computerlinguistik. Die Spracherkennung ist heutzutage schon zunehmend in den Alltag integriert: Ob bei einem automatisierten Telefon-Supportsystem oder dem integrierten persönlichen Assistenten im Mobiltelefon [4, S. 214-215]: Das akustische Sprachsignal enthält vor allem Informationen über den linguistischen Inhalt, aber auch Informationen über die Bedeutung, den Sprecher und die akustischen Bedingungen. Aufgabe der Sprachverarbeitung ist es den linguistischen Inhalt zu extrahieren. Bei diesem Prozess werden unterschiedlichste Schritte durchlaufen: Signalanalyse (Digitalisierung), Untereinheitenvergleich (unit-matching), lexikalische Dekodierung, syntaktische Analyse, semantische und pragmatische Analyse. Schlussendlich wird die gesprochene Sprache mit Hilfe von Hidden-Markov-Modellen (HMM) modelliert. Mehr dazu findet sich in [4, Kap. 3].

Morphologie

Die *Morphologie* (Formenlehre) beschäftigt sich mit den systematischen Beziehungen zwischen Wörtern und Wortformen und den Regeln, nach denen

Wörter und Wortformen gebildet werden. Im Zentrum stehen die *Morpheme*, welche die kleinsten bedeutungs- und funktiontragenden Elemente einer Sprache darstellen [4, S. 236]. Mehr dazu findet sich in [4, Kap. 3].

Flache Satzverarbeitung

Digitale Textdaten bilden das Ausgangsmaterial für die linguistische Verarbeitung. Im ersten Schritt der *Tokenisierung*, welche die Grundlage für sämtliche weitere Analysen darstellt, wird der Strom an digitalen Zeichen in sprachlich relevante Einheiten wie z. B. Wörter, Phrasen oder Sätze gegliedert. Dieser Segmentierung schließt sich die Annotation des linguistischen Materials an: Beim *Wortart-Tagging* (engl. Part-of-speech Tagging) werden die Wörter anschließend gemäß ihrer grammatikalischen Wortarten eingegliedert. In einem weiteren Schritt werden die einzelnen Token mit einem *Chunk-Parser* zu phrasalen Strukturen zusammengefügt. Der mit Annotationen angereicherte Text stellt nun eine wertvolle Basis für alle höhergeordneten computerlinguistischen Anwendungen dar [4, S. 264]. Mehr dazu findet sich in [4, Kap. 3].

Syntax und Parsing

Der Bereich der Syntax und des Parsing beschäftigt sich damit, wie syntaktische Strukturen repräsentiert und verarbeitet werden können. Die syntaktische Analyse von natürlicher Sprache ist in vielfältigen Anwendungsbereichen von Relevanz. Vor allem bei der maschinellen Übersetzung nimmt sie eine entscheidende Rolle bei der Disambiguierung mehrdeutiger (ambiguer) Ausdrücke ein. Zudem stellt die syntaktische Analyse die Grundlage für viele semantische und pragmatische Analysekomponenten dar. Üblicherweise werden Strukturbäume zur Notation von syntaktischen Strukturen aus der Graphentheorie (2.1.1) verwendet. Das Basisinstrument der syntaktischen Beschreibung ist die kontextfreie Grammatik. In modernen Applikationen werden aber zunehmend auch unifikationsbasierte Formalismen wie etwa PATR-II verwendet [4, S. 280-281]. Mehr dazu findet sich in [4, Kap. 3].

Semantik und Pragmatik

Die Semantik beschäftigt sich mit der Bedeutung von natürlichsprachlichen Ausdrücken von unterschiedlichen Einheiten der Computerlinguistik: Worte (lexikalische Semantik), Sätze (Satzsemantik) und Texte (Diskurssemantik). Festzuhalten ist dabei, dass sich die Semantik rein mit der wörtlichen Bedeutung von Ausdrücken beschäftigt, nicht aber mit der Handlungsbezogenheit von sprachlichen Äußerungen, so kann ein Satz auf verschiedene Arten verstanden werden und vielfältige Bedeutungen haben. Diese Aufgabe kommt

der *Pragmatik* zu [4, S. 330-331]. Eines der Kernprobleme der *Computersemantik* ist die Verarbeitung von *Ambiguitäten* (Mehrdeutigkeiten). Mehr dazu findet sich in [4, Kap. 3].

2.1.3 Ressourcen

Korpora

Aus rein theoretischer Sicht ist ein linguistisches Korpus eine Sammlung von digitalen audiovisuellen oder schriftlichen Äußerungen, welche in einer maschinenlesbaren Form vorliegen und ausschließlich authentische Sprachdaten für linguistische Zwecke beinhalten. Ein Korpus besteht aus drei Schichten: den rohen Sprachdaten, den analysierenden Annotationen und den beschreibenden Metadaten. In der Praxis wird meist jeder Text oder jede Sammlung von Dokumenten als Korpus bezeichnet, wenn diese für computerlinguistische Aufgaben genutzt werden. Dies inkludiert auch Texte die in ihrer Rohfassung als HTML Dokumente oder in XML-basierten Formaten vorliegen [4, S. 482-483]. Mehr dazu findet sich in [4, Kap. 4].

Lexikalisch-semantische Ressourcen

Zu den lexikalisch-semantischen Ressourcen zählen unter anderem Wortnetze. Die häufigsten und wichtigsten Wörtern einer Sprache sowie ihre wichtigsten Beziehungen zu anderen Wörter der Sprache werden in solchen Wortnetzen abgebildet. Semantische Verknüpfungen modellieren dabei unter anderem die Beziehung zwischen Ober- und Unterbegriffen (z.B. Sitzmöbel -> Stuhl -> Klappstuhl, Drehstuhl, Kinderstuhl) Das bekannteste Wortnetz für die deutsche Sprache ist das *GermaNet*[1]. Der Aufbau ist an die Strukturierungsprinzipien des Princeton WordNet 1.5 angelehnt, welches als „Mutter aller Netze“ gilt [4, S. 505]. Mehr dazu findet sich in [4, Kap. 4].

Sprachdatenbanken

Sprachdatenbanken sind Ausgangspunkt für die Entwicklung von Sprachtechnologie und sprachverarbeitenden Anwendungen. Eine Sprachdatenbank ist eine wohlstrukturierte Sammlung von Textdaten und Sprachsignalen in digitaler Form. Sie besteht aus unveränderlichen Primärdaten (Signaldaten) und veränderlichen Sekundär- (Annotationen) und Tertiärdaten (Lexika, Metadaten, etc.) [4, S. 524]. Mehr dazu findet sich in [4, Kap. 4].

Nicht-sprachliches Wissen

In den Bereich des nicht-sprachlichen Wissens fällt im Rahmen der Computerlinguistik vor allem das Wissen über die Welt (sog. „Weltwissen“). Detail-

[1] http://www.sfs.uni-tuebingen.de/lsd/

lierter umfasst es sowohl Faktenwissen und episodisches Wissen als auch das konzeptuelle Wissen darüber, wie unsere Auffassung der Welt strukturiert ist. Für Anwendungen der Computerlinguistik ist nicht-sprachliches Wissen essentiell, um ein komplexes *Textverstehen* modellieren zu können. Diese Wissensresourcen werden größtenteils als sogenannte *Ontologie* modelliert [4, Kap. 4]. Nicht-sprachliches Wissen stellt eine fundamentale Basis für die Kernaspekte dieser Arbeit dar und werden daher im Verlauf dieser Arbeit noch näher erläutert. Mehr zur Theorie findet sich in [4, Kap. 4].

2.2 Informationsextraktion

Die rasante Verbreitung des Mediums Internet in Kombination mit Technologien, welche die Erzeugung von benutzergenerierten Inhalten ermöglichen, haben in den letzten Jahren zu einem enormen Wachstum an frei verfügbaren Daten und Inhalten geführt. Neben all den Vorteilen und Möglichkeiten, welche diese Daten bieten, hat sich jedoch schnell eine Informationsüberflutung abgezeichnet, welche den Wunsch nach einfacheren Extraktionsmechanismen und kompakteren Repräsentationsformen hervorgerufen haben: Je mehr Inhalte online zur Verfügung stehen, desto schwieriger wird es, das Informationspotential des World Wide Web gezielt zu nutzen.

Viele Bereiche und Services der Informationstechnologie haben sich in den letzten Jahrzehnten dieses Problems angenommen. Die bekanntesten Vertreter dieser Form sind Internet Suchmaschinen wie *Google*, *Bing* und *Yahoo!*. Aber auch die Bemühungen einer Begrüdung eines *Semantic Webs* haben das Ziel Informationen in strukturierter, sinngemäß verknüpfter und in maschinenlesbarer Form aufzubereiten. Ein Forschungsgebiet, das sich in diesem Rahmen entwickelt hat, ist die *Informationsextraktion* (IE).

2.2.1 Grundlagen

Seine Ursprünge hat die Informationsextraktion in der auf Templates basierenden Extraktion von Informationen aus unstrukturierten Daten wie Fließtexten [15, S.1]: Dieser Ansatz wurde vor allem im Rahmen der *Message Understanding Conferences* der US-amerkianischen Behörde *Defence Advanced Research Agency* (DARPA) verfolgt. Dabei wurden vordefinierte Templates für sehr spezifische Aufgaben angefertigt, welche Informationen aus bestimmten Bereichen extrahieren. Ein Beispiel dafür könnte Terrorismus im Mittleren Osten sein.

Diese ersten Bemühungen automatisierte Informationsextraktion zu betreiben haben den Grundstein für das heutige Forschungsfeld gelegt. Dies ist auch in der im Jahre 1999 begründeten Definition der Informationsextraktion von Riloff und Lorenzen zu bemerken [19, S. 169]:

> IE systems extract domain-specific information from natural lan-

> guage text. The domain and types of information to be extracted must be defined in advance. IE systems often focus on object identification, such as references to people, places, companies and physical objects.

Informationsextraktion wird also verwendet um Informationen aus unstrukturierten Daten wie zum Beispiel Text, Bild, Video oder Audio zu gewinnen. In dieser Arbeit wird es sich bei der Form der unstrukturierten Daten um deutschsprachige Fließtext Korpora handeln. Die Art der Informationsextraktion kann vielfältig sein. Zu den grundlegendsten und für diese Arbeit am relevantesten Vertretern in der Computerlinguistik zählen die Entitätenextraktion sowie das Auffinden von Koreferenzen und Relationen.

Nicht zu verwechseln ist Informationsextraktion mit *Information Retrieval*. Beim *Information Retrieval* werden Dokumente aus großen Textsammlungen als Antwort zu definierten Keywords zurückgegeben, während bei der Informationsextraktion Fakten und strukturierte Daten aus Textsammlungen und Dokumenten gewonnen werden. IE legt den Grundstein für viele Computerlinguistik Applikationen [20, S. 7-8]: Text Mining, semantische Annotation, Question Answering, Opinion Mining, Sentiment Analyse sowie viele weitere.

2.2.2 Mustererkennung

Mustererkennung ist unter anderem die der Informationsextraktion zugrundeliegende Basis um Fakten aus unstrukturierten Daten zu extrahieren. Die Methoden basieren dabei entweder auf Wissen, welches von menschlichen Experten gesammelt und modelliert wurde oder auf automatisch durch maschinelles Lernen gesammeltem Wissen. Zentraler Punkt ist dabei die Erkennung von gemeinsamen Mustern und Merkmalen einer Kategorie um diese vom Inhalt anderer Kategorien zu unterscheiden. Die Klassifizierungsmuster sind dabei in abstrakter Hinsicht eine Zusammensetzung von Merkmalen und deren beschreibende Werte. Die Merkmale sind im Falle von Textdaten inhaltliche Strukturen. Während zu Beginn der Mustererkennung meist statistische und manuell erfasste Merkmale verwendet wurden, sind heute oft Feature Vektoren als Input von Algorithmen aus dem Bereich der Statistik oder des maschinellen Lernens im Einsatz [15, S.65]. Mehr dazu findet sich in [15, Kap. 4.1].

2.2.3 Entitätenextraktion

Die Entitätenextraktion (engl. Named Entity Recognition (NE)), im Deutschen auch als Eigennamenerkennung bekannt, hat das Ziel spezielle Ausdrücke wie Personen-, Firmen- und Produktnamen, Datums-, Zeit- und Maßausdrücke in einem unstrukturierten Text aufzufinden und zu extrahieren.

Bestehende Ansätze verwenden spezielle Eigennamenlisten und Automatengrammatiken [8, S. 5].

Es geht also im allgemeinen um die Identifizierung von Eigennamen und deren Klassifizierung in vordefinierte Kategorien, welche für den Zweck der Informationsextraktion von Relevanz sind. Die Entitätenextraktion bildet dabei die Basis für komplexere Informationsextraktionssysteme. Darauf aufbauende Applikationen sind im direkten Sinne Koreferenz und Relationssuche. Eine typische Entitätenextraktions-Pipeline beinhaltet ein Pre-Processing (Morphologische Analyse und flache Satzverarbeitung: Tokenisierung, Satz-Segmentierung, sowie Wortart-Tagging), das Auffinden der Entitäten mit Hilfe von statischen Gazetteer Listen und Automatengrammatiken. Im späteren Verlauf dieser Arbeit wird näher auf die typischen Ansätze und die Unterschiede bei der Verwendung mit externen Datenquellen eingegangen. Mehr dazu in Abschnitt 3.3.

2.2.4 Koreferenz und Relationen

Wichtig ist im Rahmen der Entitätenextraktion die Behandlung von *Referenzen* zwischen den einzelnen Eigennamen (Koreferenz). Somit soll sichergestellt werden, dass zum Beispiel die Personenentitäten Barack Obama, Präsident Obama, Obama oder „er“ in einem sinngemäß zusammengehörenden Absatz als eine und nicht als mehrere Personen erkannt werden. Im Detail kann man des Weiteren nach Entitäten Koreferenzen, Pronominal Referenzen (zwischen „er“, „sie“, etc. und Entitäten und Nominalphrasen) und Designationen Referenzen („der US amerikanische Präsident“) unterscheiden [8, S. 5].

Während sich die Koreferenzsuche mit der Auflösung von Entitätenduplikaten und deren Zusammenführung und Verknüpfung beschäftigt, wird bei der Relationssuche versucht, Verbindungen zwischen einzelnen Entitäten aus unterschiedlichen Kategorien zu finden. So kann zum Beispiel der aktuelle Standort des Präsidenten im Zusammenhang mit dem Text ermittelt werden, indem die Ortsentität „Washington D.C.“ und die Personenentität „Obama“ verknüpft wird. Genauso könnte eine Relation zwischen der Person „Steve Jobs“ und der Organisationsentität „Apple“ hergestellt werden.

Die Resultate der Relationssuche können unter anderem in Form einer Ontologie modelliert werden. Somit könnte sich durch die Analyse von einzelnen Texten oder ganzen Korpora eine automatisierte Wissenbasis aufbauen lassen.

Kapitel 3

Das World Wide Web als computerlinguistische Ressource

Das *World Wide Web* (WWW) hat sich seit dem Beginn seiner kommerziellen Nutzung im Jahre 1994 zu einer weltweit unerlässlichen Informations- und Kommunikationsstruktur entwickelt. Das Web beinhaltet heute einen enorm reichhaltigen Datenbestand, welcher täglich optimiert und erweitert wird. Die *Computerlinguistik* und die *Informationsextraktion* helfen allerdings nicht nur die steigende Quantität an Daten aufzubereiten und somit die *Informationsüberflutung* zu minimieren, sondern können sich gleichermaßen auch am Datenbestand des WWW bedienen um computerlinguistische Methoden zu optimieren. Die Vielzahl und die Reichhaltigkeit der verfügbaren Daten im Web spiegeln sich in der Vielfalt der potentiellen Verwendungsmöglichkeiten wider. Durch die rasche inhaltliche und technische Weiterentwicklung des Mediums entstehen des Weiteren ständig neue Möglichkeiten zur Verwendung dieser Daten in der Computerlinguistik. Ein Beispiel dafür ist die Verwendung des WWW als aufgabenspezifische Korpora für unterschiedliche Sprachen, Domänen und Textsorten. Eine besondere Bedeutung spielen dabei die von Benutzern generierten Inhalte, die mit Tags versehen werden und sogenannte *Folksonomien* [4, S. 544] bilden. Die Größe der Korpora erlaubt es zusätzlich auch sprachliche Phänomene wie z.B. Frequenzen von Bigramme abzuleiten. Diese Informationen können wiederum zum Beispiel für Disambiguierungsaufgaben eingesetzt werden.

Neben der Verwendung als Korpus und zur Extraktion von sprachlichen Eigenschaften kann vor allem das *Semantic Web* dazu verwendet werden, um bestehende Wissensbasen in computerlinguistischen Applikationen zu verwenden. Die durch semantische Technologien modellierten Informationen können zum einen grundlegend zur Verbesserung von bestehenden Algorithmen herangezogen werden, indem sie vorrangig statische Informationen

durch weitaus größere dynamische Datensätze ersetzen und ermöglichen zum anderen völlig neue Applikationen und Ansätze. Beim programmatischen Zugriff auf das WWW als Ressource sind allerdings effiziente Zugriffsmethoden unverzichtbar. Die Performanz ist der entscheidende Faktor für die Verwendbarkeit von semantischen Datenquellen und APIs in der Computerlinguistik.

In diesem Kapitel soll zuerst ein Überblick über bestehende Ansätze und deren Erläuterung in Related Work (3.1) gegeben werden, um anschließend näher auf die Verwendung von semantischen Datenquellen und APIs einzugehen. Nachfolgend werden zwei spezielle Ansätze zur Entitätenextraktion und zur Relevanzbewertung im Detail behandelt.

3.1 Related Work

Lapta und Keller geben in ihrer 2005 veröffentlichten Arbeit „Web-based models for natural language processing“ [9] zum ersten Mal einen Überblick darüber, wie das Web als Ressource in der Computerlinguistik eingesetzt werden kann. Zuvor stand hauptsächlich die Erfassung von Frequenzen von Bigrammen im Vordergrund. In ihrer Arbeit gehen Lapta und Keller über diese Möglichkeit hinaus und betrachten die Verwendung von web-basierten Inhalten in computerlinguistischen Applikationen wie beispielsweise maschinelle Übersetzung, Entdeckung von semantischen Relationen, Auflösung von Ambiguitäten und die Beantwortung von natürlichsprachlichen Fragestellungen. Als Ausgangsbasis werden Informationen zur Häufigkeit von N-Grammen herangezogen, welche über die Google Search API[1] (Deprecated) anhand der zurückgegeben Suchresultate und Ergebnisseiten verwendet. Zum Vergleich mit statischen Ressourcen wurde der im Vergleich zum Web viel kleinere BNC Korpus[2] herangezogen, welcher zum Zeitpunkt des Verfassens der Arbeit ungefähr 100 Millionen Wörter beinhaltet hat.

Die Ergebnisse der linguistischen Performanzbewertung sind in Tabelle 3.1 illustriert. Es wurde Performanz von web-basierten nicht überwachten Verfahren mit Baselinewerten (Base), BNC als Korpus (BNC) sowie dem besten Modell in der Literatur (Lit) verglichen. Bei einigen Verfahren, welche mit * gekennzeichnet sind, handelt es sich auch um Interpolationen zwischen web-basierten und korpusbasierten Methoden. Es wurden sowohl computerlinguistische Verfahren zum Erstellen von Inhalten (Generation) als auch Verfahren zur Analyse von Inhalten (Analysis) zur Evaluierung herangezogen. Es ist ersichtlich, dass alle web-basierten Verfahren eine bessere Performanz als die Baselinewerte aufweisen. Die Verfahren zur Erstellung von Inhalten performen bis auf eine Ausnahme (MT candidate selection) ebenfalls besser als die auf dem BNC Korpus basierten Methoden. Im Vergleich zu den State-of-the-art Verfahren der Literatur zeichnet sich ein gemischtes Bild

[1] https://developers.google.com/web-search/
[2] http://www.natcorp.ox.ac.uk/

Tabelle 3.1: Übersicht der Ergebnisse zur linguistischen Performanzbewertung von web-basierenden nicht überwachten Verfahren im Vergleich zu Baselinewerten (Base), BNC als Korpus (BNC), Bestes Modell in der Literatur (Lit). Die Bewertungssymbole stehen für $\triangle$: signifikant besser, $\equiv$: nicht signifikant unterschiedlich, $\triangledown$: signifikant schlechter. * Interpolationsverfahren zwischen web-basierten und korpus-basierten Methoden [9, S. 24].

Task	Ling	Type	Base	BNC	Lit
MT candidate selection *	Sem	Generation	$\triangle$	$\equiv$	$\triangle/\equiv$
Spelling correction	Syn/Sem	Generation	$\triangle$	$\triangle$	$\triangledown$
Adjective ordering *	Sem	Generation	$\triangle$	$\triangle$	$\equiv$
Article generation	Sem	Generation	$\triangle$	$\triangle$	$\triangle$
Compound bracketing	Syn	Analysis	$\triangle$	$\equiv$	$\equiv$
Compound interpretation	Sem	Analysis	$\triangle$	$\triangle$	$\triangle$
Countability detection	Sem	Analysis	$\triangle$	$\equiv$	$\triangledown$
PP attachement	Syn/Sem	Analysis	$\triangle$	$\equiv$	$\triangledown$

ab: Bei einigen Verfahren werden signifikant bessere und bei anderen wiederum signifikant schlechtere Ergebnisse erzielt. Bei Betrachtung der Verfahren zur Analyse sieht man, dass die Ergebnisse der web-basierten Analyse im Durchschnitt nicht signifikant besser sind als jene der BNC Korpus basierten Verfahren. Der Grund dafür ist, dass die Größe des Korpus hier keinen Unterschied macht, da beide Datenquellen nicht für diese Einsatzzwecke optimiert sind. Ansätze in der Literatur verwenden dafür speziell optimierte Verfahren, welche über die Verwendung von Korpora und Wörtern hinausgehen und oft für den einzelnen Einsatzzweck optimierte Baumbanken verwenden [9, S. 24].

In den letzten Jahren zeichneten sich zusätzlich auch Einsätze des *Social Semantic Web* [3, S.269] in der Computerlinguistik ab. Typische Beispiele sind die Verwendung von *kollaborativen Wissensdatenbanken* wie *Wikipedia*[3] oder *Wiktionary*[4]. Einträge im *Wiktionary* beinhalten ein breites Spektrum an lexikalischer und semantischer Information wie Wortart, Wortbedeutung, Kollokationen, abgeleitete Begriffe und Hinweise zum Sprachgebrauch sowie lexikalisch oder semantisch verwandte Begriffe verschiedener Art [4, S. 549]. Die *Wikipedia* bietet eine Vielzahl von lexikalisch-semantischen Informationen über Inhalte. Diese sind unter anderem Artikel, Graphen, semantische Tags, Thesauren, Disambiguierungen und Bedeutungsvokabulare.

Mendes et al beschreiben in ihrer Arbeit „DBpedia Spotlight: Shedding Light on the Web of Dokuments“ [14] einen ausschließlich auf *DBpedia*[5] basierenden Ansatz zur *Entitätenextraktion. DBpedia*[6] stellt basierend auf dem

[3] http://www.wikipedia.org
[4] http://www.wiktionary.org
[5] http://www.dbpedia.org
[6] http://dbpedia.org/About

Linked Open Data Paradigma die Informationen von *Wikipedia* in strukturierter und maschinenlesbarer Form auf Basis des *Semantic Web* frei zur Verfügung. DBpedia Spotlight ist eine auf DBpedia basierende Webapplikation zur Annotation von DBpedia URIs in Fließtexten. Zur Annotation werden vielfältige Informationen aus der DBpedia verwendet. Zusätzlich kann der User den Algorithmus durch diverse Treshold Anpassungen von Konfidenz- und Prominenzwerten steuern.

Der Implementierungsansatz gliedert sich in vier Schritte [14]: Die Entdeckungsphase identifiziert potentielle Phrasen, welche eine DBpedia Ressource darstellen. Die nachfolgende Kandidaten Auswahl liefert mögliche Ambiguitäten für die jeweilige Phrase. Im nächsten Schritt, dem Disambiguierungsschritt wird aus den zur Auswahl stehenden Ambiguitäten der am besten passende Kandidat ausgewählt. Verwendet werden Artikel-Label, Weiterleitungen und Informationen über Disambiguierung von DBpedia. Label sind die von Usern akzeptierte und gewählte Form der passendsten Überschrift für einen Artikel. Weiterleitungen weisen auf Synonyme, alternative Schreibweisen, Rechtschreibfehler und Akronyme hin. Disambiguierungen zeichnen mehrdeutige Begriffe aus, also gleiche Label welche unter Betrachtung des Bezugs völlig unterschiedliche Bedeutungen haben (z.B. der Begriff Washington kann für folgende Ressourcen stehen: *dbpedia:Washington,_D.C., dbpedia:George_Washington oder dbpedia:Washington_(U.S._State)*). Zur Auflösung von Ambiguität wird die Inverse Candidate Frequency (ICF) verwendet, welche die Relevanz einer Disambiguierungs-Ressource basierend auf dem unmittelbaren Umfeld und Kontext einer Entität berechnen soll.

$$ICF(w_j) = log\frac{|R_s|}{n(w_j)} = log|R_s| - log\ n(w_j) \tag{3.1}$$

Die in Gl. 3.1 illustrierte ICF geht davon aus, dass die Trennschärfe eines Wortes umgekehrt proportional zur Anzahl der damit verbundenen DBpedia Ressourcen ist. Sei R_s das Set an potentiellen Kandidaten für eine Phrase s und $n(w_j)$ die Anzahl der Ressourcen in R_s, welche mit dem Wort w_j assoziiert sind.

Sämtliche DBpedia Ressourcen werden anschließend in einem Vector Space Model (VSM) abgebildet und mit Term Frequency (TF) und ICF Werten versehen. Die Auflösung der Ambiguitäten erfolgt durch das Lösen eines Ranglistenproblems eines berechneten Ähnlichkeitswerts (Similarity Score) zwischen den Kontext- und den Umgebungsvektoren. Sämtliche Variablen der Berechnungen lassen sich in der Benutzeroberfläche anpassen. Dazu zählen unter anderem *Ressource Sets to Annotate*, wo sich die Kategorien und Themengebiete der zu findenden Entitäten festlegen lassen. *Resource Prominence*, ein Treshold Wert für die Prominenz (berechnet aus der Anzahl der einkommenden Wikipedia Links), *Topic Pertinence*, ein Treshold Wert für die Kontextrelevanz von Disambiguierungen, *Contextual Ambiguity*, welche beschreibt, wie mehrdeutig der umgebende Kontext ist und *Disambiguation*

Tabelle 3.2: F-Maß der unterschiedlichen Systeme im Vergleich [14].

System	*F-Maß*
The Wiki Machine	59,5%
DBpedia Spotlight (Beste Konfiguration)	56,0%
DBpedia Spotlight (Keine Konfiguration)	45,2%
Zemanta	39,1%
Open Calais+ Naive	16,7%
Alchemy	14,7%
Ontos+ Naive	6,7%
Open Calais	6,7%
Ontos	1,5%

Confidence, ein Wert zwischen null und eins, welcher den Konfidenzgrad der Entscheidung bei der Auflösung von Ambiguitäten beschreibt.

Im Vergleich zu Verfahren von anderen Systemen wie *OpenCalais*[7], *Zemanta*[8], *Ontos Semantic API*[9] , *The Wiki Machine*[10] und *Alchemie API*[11] zeichnete sich ab, dass das F-Maß von DBpedia Spotlight, welches Genauigkeit (Precision) und Trefferquote (Recall) vereint, stark von der gewählten Konfiguration abhängig ist. Wie in Abbildung 3.1 dargestellt wird, schwanken die Recall und Precision Werte je nach Konfigurationsstufe. Die einzelnen Punkte in den Linien der Konfigurationsstufen sind der Konfidenzgrad (0.1 bis 0.9).

Das aus den Recall und Precision Werten berechnete F-Maß beschreibt die Genauigkeit des Algorithmus. In Tabelle 3.2 werden die errechneten F-Maße miteinander verglichen.

Die Ergebnisse zeigen sehr deutlich, dass durch eine manuelle Anpassung der Konfiguration das F-Maß deutlich gesteigert werden kann: von 45,2% ohne Konfiguration auf 56% mit der besten Konfiguration. Dies zeigt auch, dass die Konfigurationen und Ergebnisse immer stark vom jeweiligen Korpus abhängen können.

Bosca et al beschreiben in ihrer Arbeit „Automatic Gazetteer Generation from Wikipedia“ [2] einen ähnlichen Ansatz zur automatisierten Erstellung von multilingualen Gazetteer Listen aus DBpedia. Der Schwerpunkt dieser Arbeit liegt allerdings in der Klassifizierung von DBpedia Ressourcen in für die Entitätenextraktion relevante Kategorien wie Person, Ort, Organisation und Beruf. Der Ansatz beschränkt sich allerdings auf einen Bruchteil der in Wikipedia verfügbaren Ressourcen und betrachtet ausschließlich

[7] http://www.opencalais.com
[8] http://www.zemanta.com
[9] http://www.ontos.com
[10] http://thewikimachine.fbk.eu
[11] http://www.alchemyapi.com/

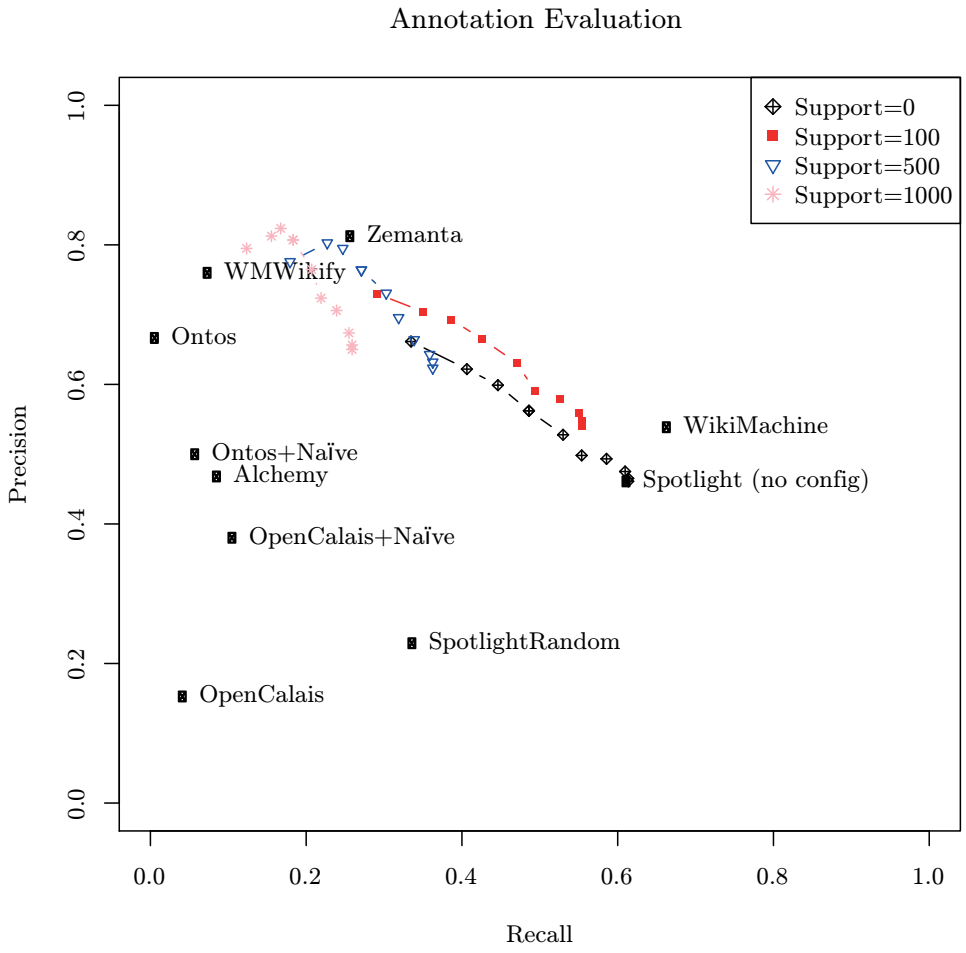

Abbildung 3.1: Unterschiedliche Konfigurationen von DBpedia Spotlight im Vergleich mit anderen Systemen [14].

Entitäten in englischer Sprache. Zur Klassifizierung werden die bestehenden Wikipedia Kategorien verwendet und daraus die passendste Kategorie abgeleitet. Das Ergebnis zeigt ein äußerst zufriedenstellendes Resultat mit einem durchschnittlichen F-Maß von 0,93 und einer korrekten Klassifizierung von 95,6%. Die Übersicht der Precision, Recall sowie F-Maß Werte findet sich in Tabelle 3.3.

Die Relevanzbewertung auf Entitätenebene ist eine sehr neue noch großteils unerforschte Domäne. Während das Ranking von Dokumenten oder Websites nicht zuletzt seit der Einführung von Suchmaschinen stark an Popularität gewonnen hat, wurde die Relevanz von einzelnen Entitäten noch kaum erforscht. Es gibt jedoch einige Arbeiten, welche sich damit befassen, einzelne Entitäten im Zusammenhang mit Dokumenten zu sehen. Vercoustre et al beschäftigen sich zum Beispiel in ihrer Arbeit „Entity Ranking

Tabelle 3.3: F-Maß der Klassifizierung pro Entitäten Kategorie [2].

Precision	*Recall*	*F-Measure*	*Class*
0,931	0,952	0,941	Person
0,985	0,983	0,984	Location
0,889	0,847	0,867	Organization
0,954	0,944	0,949	Work

in Wikipedia“ mit der Relevanzbewertung von einzelnen Wikipedia Seiten [22]. Einen ähnlichen Ansatz verfolgen Zaragoza et al in ihrer Arbeit „Ranking Very Many Typet Entities on Wikipedia“ [24]. Demartini at al untersuchen die Verwendung von einzelnen Entitäten im Rahmen von Such algorithmen [6].

3.2 Datenquellen

Die Anzahl und Vielfalt an verfügbaren Daten im Web ist gerade mit dem Aufkommen des *Social Web* [3, Kap. 3] enorm gestiegen. Ständig entstehen neue Services, welche das Erstellen von benutzergenerierten Inhalten ermöglichen. Die wichtigsten Kategorien sind Social Networks (z.B. Facebook, Google+), Diskussion (z.B. Blogs, Foren, Microblogging), Wissensaustausch (z.B. Wikis), Social News (z.B. Reddit), Social Bookmarking (z.B. Delicious), Multimedia Sharing (z.B. Instagram, Pinterest, Viddy). In diesen und weiteren Kategorien entsteht jede Sekunde eine enorme Menge an neuen Daten.

Nicht alle diese Daten sind allerdings in der vorliegenden Form brauchbar. Aus diesem Grund gibt es eine Vielzahl an Bemühungen, die versuchen die Qualität der Daten zu steigern, die Daten sinnvoll zu verknüpfen und maschinenlesbar zugänglich zu machen. Diese Bemühungen werden unter dem Begriff des *Semantic Web* [3, Kap. 4] zusammengefasst. Tim Berners-Lee, der Begründer des World Wide Webs hat auch den Begriff des Semantic Web geprägt und definiert seine Vision wie folgt [1, Kap. 12]:

> I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A 'Semantic Web', which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The 'intelligent agents' people have touted for ages will finally materialize.

Diese Vision hat sich in den letzten Jahren allerdings nur sehr marginal durchgesetzt. Nichtsdestotrotz gab es in den letzten Jahren eine Vielzahl

an Bemühungen vor allem im Rahmen des *Linked Open Data*[12], bei der reichhaltige Informationen über semantische Technologien unter Verwendung von URIs[13] und RDF[14] zugänglich gemacht wurden.

Neben den vom W3C[15] spezifizierten semantischen Technologien finden Application Programming Interfaces (APIs)[16] zunehmend an Verbreitung. Die Gründe dafür sind praktischer Natur: Zum einen ermöglicht es der Zugang zu einer Plattform über eine API Dritt-Entwickler miteinzubeziehen und somit für mehr Innovation und Verbreitung zu sorgen. Zum anderen sind APIs heutzutage fast unumgänglich um die Daten der eigenen Anwendung auf unterschiedlichen Endgeräten wie Smartphones zugänglich zu machen. Beide Formen ermöglichen den Zugang zu Daten und den Einsatz in eigenen Applikationen auf unterschiedliche Art und Weise. Im Nachfolgenden werden sowohl ausgewählte semantische Datenquellen als auch APIs, im Speziellen jene aus dem Social Web, in Hinsicht auf deren Verwendung in der Computerlinguistik betrachtet.

3.2.1 Semantische Datenquellen

Grundlagen

Dem *Semantic Web* liegt die Idee der Wissensrepräsentation zu Grunde: es sollen unstrukturierte Daten in ein Vokabular überführt werden, welches bestimmte Strukturen befolgt und Verbindungen modelliert. Es sollen also Informationen, welche in menschlicher Sprache ausgedrückt wurden, mit einer eindeutigen Beschreibung und Semantik (Bedeutung) versehen werden, denn eine Verarbeitung durch Maschinen ist nur dann möglich, wenn Dinge eindeutig zugeordnet werden können und Verbindungen von diesen verstanden werden können.

Zur Modellierung dieses Wissens wird üblicherweise die Ontologie-Repräsentationssprache RDF[17] oder die Web Ontology Language OWL verwendet. Wissen wird dabei in Form von Verbindungen über sogenannte RDF-Triple abgebildet. Ein RDF-Triple setzt sich aus Subjekt, Prädikat und Objekt zusammen.[18] Siehe Abbildung 3.2. Jede einzelne Ressource ist über eine einzigartige URI[19] und einen zugeordneten Namespace[20] zugänglich.

Die Summe der RDF-Triple beschreibt ein Vokabular oder bei komplexeren und größeren Konstrukten eine Ontologie[21]. Die graph-basierte Abfrage-

[12] http://www.linkeddata.org
[13] http://en.wikipedia.org/wiki/URI
[14] http://en.wikipedia.org/wiki/Resource_Description_Framework
[15] http://www.w3.org
[16] http://en.wikipedia.org/wiki/Application_programming_interface
[17] http://en.wikipedia.org/wiki/Resource_Description_Framework
[18] http://www.w3.org/TR/rdf-concepts/#section-triples
[19] http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/#dfn-URI-reference
[20] http://www.w3.org/TR/REC-rdf-syntax/#section-Namespace
[21] http://www.w3.org/standards/semanticweb/ontology

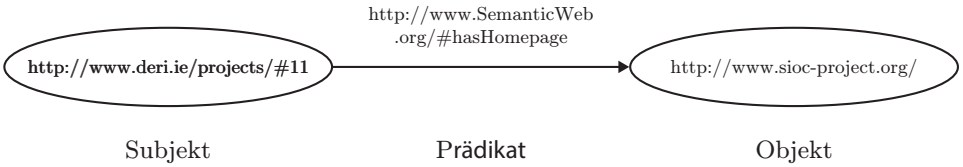

Abbildung 3.2: Simple Darstellung eines RDF Triple: Zwei Knoten (Subjekt und Objekt) und ein gerichteter Graph (Prädikat) [3, S. 53]

sprache *SPARQL* (SPARQL Protocol And RDF Query Lanuage)[22] erlaubt es anschließend vielfältige Abfragen auf die Datenbasis auszuführen. Mehr zu den Grundlagen und Konzepten des Semantic Web findet sich in [27].

Bei Betrachtung der Visualisierung des Semantic Webs, welche von der *Linked Open Data* Community[23] zuletzt im September 2011 erstellt wurde, lässt sich schnell erkennen, dass es sich dabei nach wie vor um eine überblickbare Anzahl an Plattformen handelt und dass die beiden Ontologien DBpedia[24] und Freebase[25] einen bedeutenden Anteil der Verbindungen modellieren und sich deshalb im Zentrum der Visualisierung befinden. Beide Ontologien beinhalten knapp über 20 Millionen Datensätze[26].

Verwendung in der Computerlinguistik

Semantische Technologien und Ontologien lassen sich nicht nur innerhalb der Computerlinguistik zum Abspeichern von gewonnenem Wissen und zum Aufbau von Wissensbasen über Korpora verwenden, sondern auch der Einsatz von externen semantischen Datenquellen kann in vielen Anwendungen sehr hilfreich sein. So kann zum Beispiel das in Ontologien modellierte Wissen dazu verwendet werden, um in natürlich sprachlicher Form vorliegende Fragen zu beantworten um somit z.B. „Frage und Antwort“[27] Aufgaben zu lösen. Ein weiterer möglicher Einsatz ist die Verwendung von Ontologien um semantische Prozesse in der Computerlinguistik zu optimieren um Sachverhalte besser zu verstehen und um Fehlinterpretationen zu vermeiden. Ein Beispiel dafür sind die Informationen über Ambiguitäten (Mehrdeutigkeiten) in DBpedia, welche die Möglichkeit schaffen, mehrdeutige Begriffe eindeutig zuordnen zu können. Ein weiterer Ansatz, welcher in dieser Arbeit näher verfolgt wird, ist die Entitätenextraktion basierend auf semantischen Datenquellen. Mehr dazu im Abschnitt 3.3.

[22] http://www.w3.org/TR/rdf-sparql-query/
[23] http://linkeddata.org/
[24] http://www.dbpedia.org
[25] http://www.freebase.com
[26] http://wiki.dbpedia.org/Datasets
[27] http://en.wikipedia.org/wiki/Question_answering

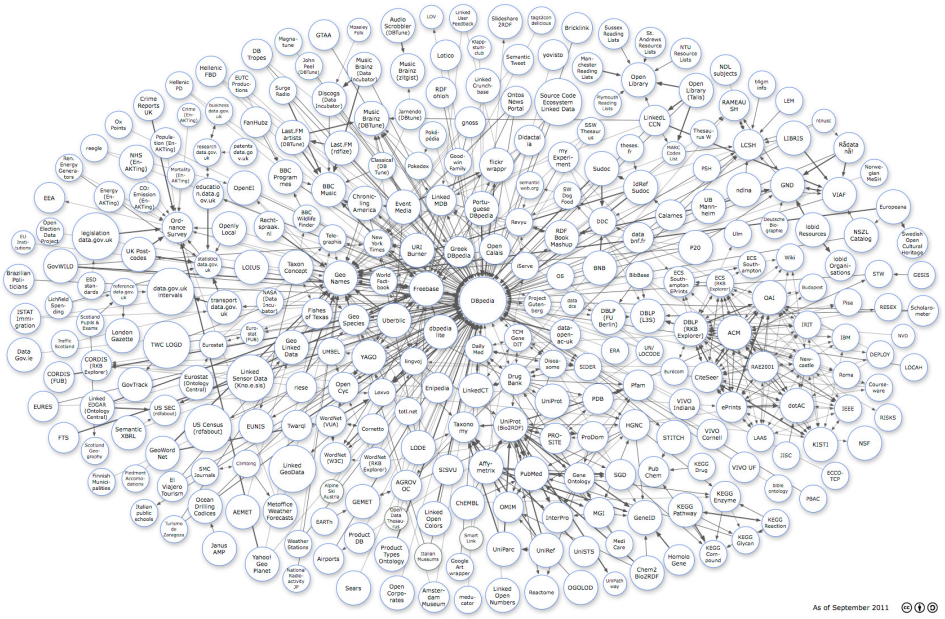

Abbildung 3.3: Die *Linked Open Data Cloud* visualisiert die wichtigsten Bestandteile und Verbindungen im Semantic Web. Letztes Update im September 2011 [26].

DBpedia

Wikipedia wurde im Jahr 2011 online gestellt und ist heute eine der meistbesuchten Websites und die weltweit größte Enzyklopädie. Zum Zeitpunkt des Verfassens dieser Arbeit hat Wikipedia 23 Millionen Artikel in 275 aktiven Sprachausgaben, davon 4 Millionen in der englischen Ausgabe[28]. *DBpedia* hat es sich zur Aufgabe gemacht, diese reichhaltige Informationsquelle in strukturierter und maschinenlesbarer Form auf Basis des *Semantic Web* zur Verfügung zu stellen.

Die Vielfalt und Reichhaltigkeit der Inhalte wurde durch einen einzigartigen, offenen und kollaborativen Prozess geschaffen, welcher heute unter dem Begriff der *Wiki Software* zusammengefasst wird und auf viele andere Bereiche u.a. im Wissensmanagement übertragen wurde. Eine Vielzahl dieser Daten kann für die Anwendung in der Computerlinguistik von größter Bedeutung sein. Im Nachfolgenden sollen die wichtigsten Datenstrukturen für diesen Einsatz erläutert werden [13, S. 718 - 732].

Artikel Ein Artikel stellt die Einheit an Information in Wikipedia dar. Jeder Artikel besteht aus einem kurzen und bündigen Titel, einer knappen

[28] http://en.wikipedia.org/wiki/Wikipedia

Abbildung 3.4: Deutscher Wikipedia Artikel über Wikipedia.

Kurzfassung, einem übersichtlichen Inhaltsverzeichnis, dem eigentlichen Inhalt, einer Infobox mit Fakten sowie einem Literaturnachweis. Jeder Artikel beschäftigt sich mit einem einzigartigen Konzept, welches im ersten Satz eines Artikels beschrieben wird. Die Verwendung der Titel kann in der Computerlinguistik zur *Entitätenextraktion* verwendet werden. Mehr dazu im Abschnitt 3.3. Die Fakten in der Infobox können wiederum zur Anreicherung von Informationen zu bestehenden Entitäten dienen.

Disambiguierungs-Seiten Anstatt User, welche nach einem bestimmten Begriff suchen, direkt auf einen Artikel zu führen, werden diese oft auf Disambiguierungs-Seiten geleitet. Solche Seiten zeichnen sich durch den Zusatz „_(Begriffsklärung)“ oder „_(disambiguation)“ im Englischen am Ende des Titel aus und listen unterschiedliche Bedeutungen von Wörtern und Namen geordnet nach Kategorien auf.

Das Auflösen von Ambiguitäten ist eine der größten Herausforderungen in der Computerlinguistik, gerade bei der Erkennung von Entitäten. Die Informationen aus Wikipedia liefern dabei vollständige und qualitativ hochwertige Listen an potentiellen Bedeutungen für einen Begriff. Durch die vorhanden

Zusatzinformationen und Fakten aus der Infobox sowie eine Betrachtung des umliegenden Inhalts im vorkommenden Text, können diese Informationen bei einer korrekten Zuweisung helfen.

Weiterleitungen Während Disambiguierungen unterschiedliche Artikel zu einem Begriff auflisten, kümmern sich Weiterleitungen darum, unterschiedliche Namen, Schreibweisen, Tippfehler, Synonyme und sonstige Variationen, welche alle ein gemeinsames Konzept beschreiben, zum korrekten Artikel weiterzuleiten.

Auch Weiterleitungen haben eine große Bedeutung in der Computerlinguistik, während mit statischen Datenquellen und Regeln nur eine begrenzte Anzahl an unterschiedlichen Schreibweisen unterstützt werden kann, sind diese Daten viel reichhaltiger. Auch Tippfehler waren in Anwendungen der Computerlinguistik lange ein Problem bei der semantischen Interpretation. Mit den von der Wikipedia Suche gesammelten Daten können so auch Entitäten mit Tipp- oder Rechtschreibfehlern mit relativ geringem Aufwand korrekt erkannt werden.

Hyperlinks Wikipedia ist nach dem Prinzip der Quernavigation aufgebaut. Es gibt also keine durchgehende Navigation sondern eine Vielzahl an Hyperlinks innerhalb der Inhalte, welche Querverweise zu anderen Artikeln enthalten. Ein durchschnittlicher Wikipedia Artikel enthält mehr als 25 Hyperlinks[13].

Die Anzahl der eingehenden Hyperlinks zu einem Artikel kann Aufschluss über die Popularität und Wichtigkeit eines Artikel geben. Diese Information kann zum einen zur Bewertung der Globalrelevanz von Entitäten verwendet werden, aber auch bei Disambiguierungsaufgaben helfen. Wenn zum Beispiel für einen Algorithmus zwischen gleichwertigen Bedeutungsformen ausgewählt werden muss, wäre eine besonders simple Vorgehensweise die Wahl der Bedeutungsform mit der höheren Popularität. Hyperlinks sind eine weitere wertvolle Quelle für Synonyme, welche durch Weiterleitungen nicht abgedeckt sind. Durch die vielfältige Verwendung in unterschiedlichsten Artikeln entsteht eine Vielzahl an unterschiedlichen Schreibweisen und Namensgebungen für einen Artikel.

Kategorien Editoren von Artikeln haben die Aufgabe diese in Kategorien einzuordnen. Jede Kategorie muss des Weiteren wieder in weitere, generischere Kategorien eingeteilt werden. So entsteht ein mächtiges Geflecht an Kategorien, welche in ihrer Gesamtheit am besten als Folksonomie beschrieben werden kann, da die Kategorien, wie auch die Artikel von jedem bearbeitet werden können. Das Ziel ist es die Informationen der Wikipedia in einer hierarchischen Form abzubilden. Kategorieninformationen können zum Beispiel zur Klassifizierung von computerlinguistischen Ressourcen verwendet wer-

den. Dieser Ansatz wird zum Beispiel in Wang et al 2009 [23] näher verfolgt. Im Rahmen der Entitätenextraktion können die Kategorieninformation zur einwandfreien Entitätentypenzuweisung verwendet werden.

Infoboxen Infoboxen stellen die wohl wertvollsten Informationen für die maschinelle Weiterverarbeitung dar, da sie strukturierte und nach Kategorien uniformierte Inhalte beinhalten. Je nach Kategorien werden unterschiedliche Templates verwendet: So haben zum Beispiel alle Länder die selben Felder zur Verfügung wie Einwohneranzahl, Bevölkerungsdichte, Amtssprache, Fläche etc.

Diese Informationen können in der Computerlinguistik vor allem zur Anreicherung von Entitäten mit Fakten verwendet werden. So kann zum Beispiel das Alter oder der Geburtsort bei einer Personenentität hinzugefügt werden. Dies kann zum einen für den Benutzer der Applikation interessant sein, aber auch bei der semantischen Analyse von Inhalten helfen.

3.2.2 Social Web

Als Social Media werden alle Medien (Plattformen) verstanden, die die Nutzer über digitale Kanäle in der gegenseitigen Kommunikation und im interaktiven Austausch von Informationen unterstützen.[29] Diese Weiterentwicklung des Web hat in den letzten Jahren zu einer unvergleichbaren Euphorie und zu einer enormen Vielfalt an von Benutzern generierten Inhalten geführt. Die Qualität variiert dabei drastisch von sehr wertvollen bis zu Spam-Inhalten.

Für den Einsatz in der Computerlinguistik sind vor allem aggregierte statistische Informationen von Bedeutung. Des Weiteren sollte es sich bei der Form der von Benutzern erstellten Inhalte idealerweise um Textinformation handeln. Die bekanntesten Social Media Plattformen zur Erstellung und zum Austausch von primär textuellen Informationen sind *Facebook* und *Twitter*.

Facebook

Facebook ist das größte und global weit verbreitetste soziale Netzwerk und hatte im Juni 2012 955 Millionen aktive Benutzer, was ein Wachstum von 29% pro Jahr darstellt [7]. Das Wachstum seit der Gründung im Jahr 2004 ist in Abbildung 3.5 zu sehen.

Über die letzten Jahre hat Facebook eine Vielzahl von Informationen angesammelt. Diese Datenbasis stellt auch das wichtigste Asset von Facebook dar um die Plattform zu monetarisieren. Ein Zugriff auf diese Daten ist mit Hilfe der auf einer REST Architektur basierenden *Graph API*[30] möglich. Auf diesem Weg können beschränkte Informationen zu den unterschiedlichen Facebook Objekten und Entitäten abgefragt werden. Jedes Facebook Objekt

[29] http://de.wikipedia.org/wiki/Social_Media
[30] https://developers.facebook.com/docs/reference/api/

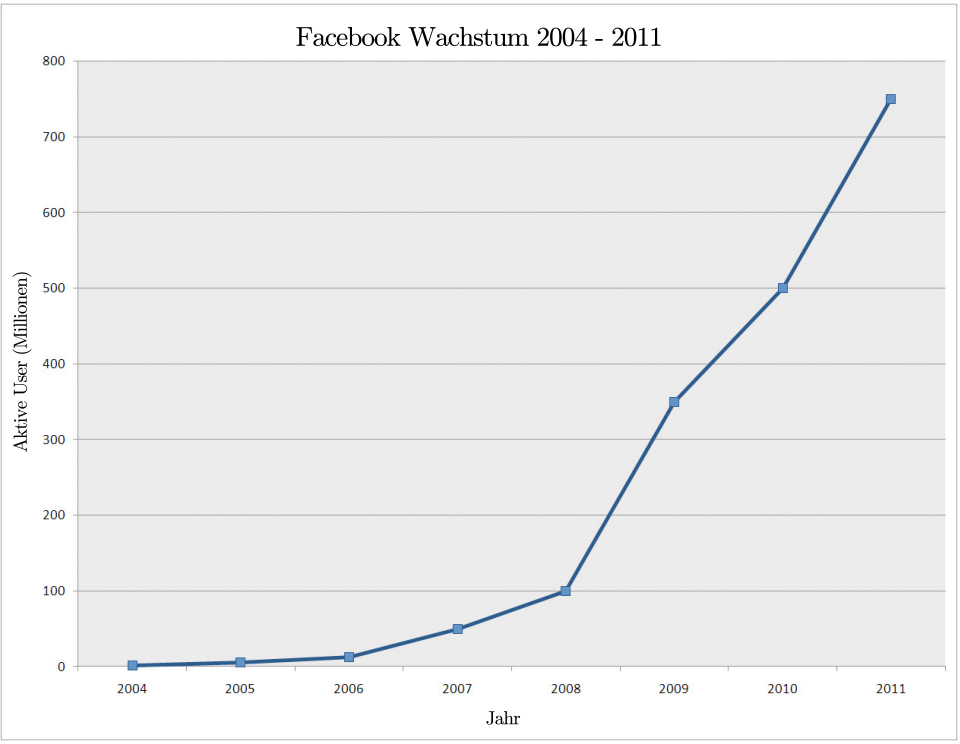

Abbildung 3.5: Aktive User auf Facebook seit 2004[7].

ist dabei mit einer eindeutigen ID gekennzeichnet. Die API ist öffentlich zugänglich und liefert Informationen im JSON Format. So gibt zum Beispiel der GET Aufruf der URI „https://graph.facebook.com/19292868552“ Informationen zur Faceboook Seite „Facebook Developers“ zurück:

```
{
   "name": "Facebook Developers",
   "is_published": true,
   "website": "http://developers.facebook.com",
   "username": "FacebookDevelopers",
   "company_overview": "Facebook Platform enables anyone to build...",
   "about": "Build and distribute amazing social apps on Facebook,
   "talking_about_count": 15965,
   "category": "Product/service",
   "id": "19292868552",
   "link": "https://www.facebook.com/FacebookDevelopers",
   "likes": 144064,
   "cover": {
      "cover_id": "10151008748223553",
      "source": "http://sphotos-a.ak.fbcdn.net/hphotos-..._n.jpg",
      "offset_y": 0
   }
}
```

Der potentielle Einsatz dieser Daten in der Computerlinguistik ist vielfältig. So können diese zum Beispiel dazu eingesetzt werden, um Textanalysen auf einer persönlichen Ebene durchzuführen. Die personenbezogenen Daten können dabei dafür verwendet werden, um zum Beispiel persönliche Interessen in eine Korpus-Analyse miteinzubeziehen um für den User relevantere Ergebnisse zu liefern.

Eine weitere Verwendung ist der Einsatz von statistischen Informationen wie zum Beispiel die Anzahl der „Gefällt mir“ oder „Sprechen darüber“ einer bestimmten Facebook Seite um die Relevanz von Entitäten zu bestimmen. Mehr dazu im Abschnitt 3.4. Ein Einsatz der von Benutzern generierten Inhalten als Korpus zum Trainieren von z.B. Klassifizierungsmechanismen ist ein weiteres potentielles Einsatzgebiet.

Twitter

Twitter ist in seinen Grundzügen ein Microblogging Dienst, welcher es seinen Benutzern erlaubt, Informationen in kurzer und prägnanter Form in Tweets mit einer Länge von maximal 140 Zeichen zu teilen. Alle Inhalte sind grundsätzlich öffentlich zugänglich, können jedoch im Nachhinein auf die Follower eingeschränkt werden. Eine Besonderheit ist die hohe Aktualität der Inhalte und die äußerst schnelle Verbreitung von Neuigkeiten. Gerade die Information über Aktualität kann für die Computerlinguistik von Bedeutung sein. Unter anderem kann diese bei der Relevanzbewertung von Entitäten eingesetzt werde. Mehr dazu im Abschnitt 3.4.

3.2.3 Google AdWords API

Google AdWords[31] ist das wichtigste Werbemodell der Suchmaschinen und somit auch die Haupteinnahmequelle.[32] Werbetreibende können dieses Service dazu verwenden um Werbeeinblendungen zu schalten im Text oder Banner Format, in den Search Engine Result Pages (SERPs) zu einem bestimmten Begriff oder auf Google Partnerseiten, welche das AdWords Werbemodell implementiert haben. Die Kosten werden dabei wahlweise nach einem cost-per-click (CPC) oder einem cost-per-mille (CPM) Model abgerechnet. Bei der Buchung von Werbeeinschaltungen werden dem Werbetreibenden eine Vielzahl von sehr wertvollen Informationen angeboten. Dazu zählen unter anderem verwandte Suchbegriffe, monatliche globale und lokale Suchanfragen sowie die durchschnittlichen cost-per-click (CPC) Werte.

Die Google AdWords API[33] stellt ein SOAP Interface zur Verfügung, welches sämtliche der obigen Daten zugänglich macht. Die Verwendung dieser API ist jedoch ausschließlich für den Einsatz im Werbemanagement gedacht

[31] adwords.google.com
[32] http://investor.google.com/financial/tables.html
[33] https://developers.google.com/adwords/api/

und nur über einen kostenpflichtigen API Key zugänglich. Die Kosten pro 1000 API Aufrufe betragen 0,25$.[34]

Nichtsdestotrotz sind diese wertvollen Daten in der Computerlinguistik sehr hilfreich: Lokale und globale monatliche Suchanfragen können zur Relevanzbewertung verwendet werden. Die verwandten Suchbegriffe erlauben des Weiteren eine Relevanzbewertung in Hinsicht auf einen bestimmten Kontext. Mehr dazu im Abschnitt 3.4.

3.2.4 Limitationen

Der Einsatz von externen Datenquellen in der Computerlinguistik kann viele Vorteile mit sich bringen und bestehende Verfahren optimieren oder Möglichkeiten für ganz neue Anwendungen schaffen, welche ohne diese nicht denkbar gewesen wären. Nicht alle inhaltlich relevanten Daten sind jedoch aufgrund von technischen Limitierungen für die Verwendung in der Computerlinguistik brauchbar.

Die erste Hürde ist, dass eine Vielzahl von Daten nicht in maschinenlesbarer Form zugänglich sind. Oft gibt es gar keine Möglichkeit um auf diese mit semantischen Technologien oder APIs zuzugreifen. In anderen Fällen gibt es Duplikate von den originalen Daten um diese im Rahmen des Semantic Web zugänglich zu machen, wie es zum Beispiel bei DBpedia der Fall ist. Hier ist die Aktualität und Synchronisierung der Inhalte oft ein Problem. Viele APIs von kommerziellen Services machen zudem nur einen sehr geringen Teil der Informationen verfügbar, da diese Daten eines der wichtigsten Güter für sie darstellen.

Eine weitere Hürde ist das Thema Performanz: In der Computerlinguistik müssen pro Dokument eine Vielzahl an Analysen durchgeführt werden. Auf ein gesamtes Korpus gesehen ist die Anzahl der Einzelanalysen noch um ein Vielfaches höher. Während Berechnungen und Auswertungen von Regeln schon eine beträchtliche Zeit benötigen – abhängig von der verfügbaren Rechenleistung und der Größe des Korpus – stellen HTTP Anfragen zu externen Services teilweise ein viel größeres Performanzproblem dar. Je nach Einsatzzweck soll dabei die Anzahl der HTTP Anfragen möglichst gering gehalten werden. Eine Möglichkeit ist dabei die Verwendung von so genannten Batch-Requests, welche das gleichzeitige Abrufen von mehreren Anfragen an eine Service API mit nur einem HTTP Request erlaubt. Die Graph API von Facebook erlaubt zum Beispiel das Absetzen von Batch-Requests mit einer maximalen Anzahl von 300 Einzelrequests pro Anfrage.[35] Bei der Google AdWords API sind sogar bis zu 2000 Anfragen in einem Batch-Request möglich je nach Typ des verwendeten Services.

Gerade bei semantischen Datenquellen gibt es eine weitere Möglichkeit ein Duplikat der Ontologie auf dem eigenen Server abzubilden, welches regel-

[34] https://developers.google.com/adwords/api/docs/ratesheet
[35] https://developers.facebook.com/docs/reference/api/batch/

mäßig mit dem Master abgeglichen wird. Die Computerlinguistik Anwendung kann in diesem Fall direkt mit dem SPARQL-Endpoint am eigenen Server kommunizieren, wobei die Abarbeitungsgeschwindigkeit somit wieder auf die CPU und den Arbeitsspeicher des eigenen Servers ausgelagert wird.

Bei der Verwendung von APIs stellen Rate Limits zudem eine bedeutende Einschränkung dar. Um die Auslastung ihrer Services in Grenzen zu halten und Kosten einzusparen limitieren viele Anbieter die Anzahl der Requests in einem gegeben Zeitraum. Ein Beispiel dafür ist die Twitter API: pro API OAuth Token können maximal 350 Abfragen pro Stunde abgesetzt werden.[36] Bei kostenpflichtigen APIs wie der Google AdWords API sind die Kosten pro Dokument möglichst gering zu halten. Der Mehrwert in der Analyse muss in einem akzeptablen Verhältnis zu den aufkommenden Kosten stehen.

Diese und weitere Einschränkungen sind bei der Verwendung von externen Datenquellen in der Computerlinguistik stets zu beachten und sollten bereits vor der Implementierung genau untersucht werden. Nur durch ausreichende Zugänglichkeit zu den Daten und vor allem durch zufriedenstellende Performanz können Computerlinguistik Applikationen für den Einsatz in der Praxis geschaffen werden. Für wissenschaftliche Zwecke und zur Evaluierung der Möglichkeiten ist jedoch ein Einsatz auch bei nicht akzeptabler Performanz denkbar.

3.3 Entitätenextraktion

Wie bereits im Abschnitt 2.2.3 erwähnt, hat die Entitätenextraktion (engl. Named Entity Recognition (NE)) – im Deutschen auch als Eigennamenerkennung bekannt – das Ziel, spezielle Ausdrücke wie Personen-, Firmen- und Produktnamen, Datums-, Zeit- und Maßausdrücke in einem unstrukturierten Text aufzufinden und zu extrahieren. Eine genaue Auflistung der Entitätentypen findet sich in Tabelle 3.4.

Im Nachfolgenden soll die Verwendung von externen Datenquellen bei der Entitätenextraktion betrachtet werden und ein spezieller Ansatz vorgestellt werden, welcher die Vorteile von herkömmlichen Herangehensweisen in Kombination mit neuen Ansätzen verbindet.

3.3.1 Herangehensweise

Bei der Betrachtung von Related Work (3.1) und existierenden Implementierungen wurde beobachtet, dass nahezu alle bestehende Ansätze spezielle Eigennamenlisten und Automatengrammatiken [8, S. 5] verwenden. Die Verwendung von externen Datenquellen ist aber ebenfalls nicht gänzlich neu. Wie zum Beispiel in Mendes et al 2012 [14] beschrieben, verwendet DBpedia

[36] https://dev.twitter.com/docs/rate-limiting

Tabelle 3.4: Liste an zu erfassende Entitäten

Entität	**Aufgabe**	**Beispiele**
Person	Erfassen von Individuen	z.B. Vorname (Andreas), Vorname Nachname (Stephan Hamberger), Titel + Name (Dr. Meier), Anrede Name (Frau Huber)
Datum	Erfassung von konkreten Jahreszahlen, Zeitangaben, Uhrzeiten, Datumsangaben, Zeitspannen	seit / vor 10 Jahren, 2012, 8:00 Uhr, 13:00-17:00, 11.06.2012, 11. Juni, September, Herbst 2012
Beruf	Erfassung von unternehmensinternen und externen Positionsbezeichnungen	CEO, Produkt Manager, Marketingleiter
Organisation	Erfassung von Firmenbezeichnungen, Organisationen, Parteien, etc.	ACTUAL, ACTUAL Fensterbau GmbH, SPÖ, Oberlandesgericht, Kleintierzuchtverein Enns
Ort	Erfassung von geografischen Bezeichnungen, wie sie z.B. in Landkarten vorkommen (Länder, Kontinente, Städte, Flüsse, Regionen, Seen, Gebirge)	Österreich, Europa, Enns, Mühlviertel, Alpen
Geldbeträge	Erfassung von numerischen Geldbeträgen in Zahl oder Wortform mit zugehörender Währungsangabe	5€, zehn Dollar

Spotlight[37] diverse Informationen von DBpedia um Wikipedia Entitäten im Text zu erkennen. Weitere ähnliche Arbeiten sind in Abschnitt 3.1 zu finden.

Nach genauerer Betrachtung der Ansätze und deren Evaluierung konnte festgestellt werden, dass sowohl herkömmliche Herangehensweisen mit statischen Gazetteer Listen als auch die Verwendung von externen Datenquellen einzelne Vorteile bieten. Aus diesem Grund wurde versucht eine Lösung zu finden, welche die Vorteile beider Lösungsansätze ineinander vereint. Nachfolgend soll ein Ansatz speziell für den Einsatz in der deutschen Sprache begutachtet werden.

3.3.2 Verwendete Datenquellen

Um die Vorteile von bestehenden Ansätzen, welche statische Listen verwenden, mit denen von neuen Methoden und dem Einsatz von externen Datenquellen zu kombinieren, wurden sowohl statische Gazetteer Listen als auch Daten von DBpedia (3.2.1) verwendet.

[37] https://github.com/dbpedia-spotlight

Statische Gazetteer Listen

Statische Gazetteer Listen sind meist manuell angelegte Sammlungen von einzelnen Listen mit Entitätennamen im *Plaintext*[38] Format. Die jeweiligen Listen sind dabei üblicherweise in einzelne Dateien ausgelagert, welche pro Zeile eine Entität beinhalten und jeweils logische Gruppen repräsentieren z.B. Namen von Städten, Organisationen, Datumsangaben, etc. Eine Index Datei fügt schließlich alle einzelnen Dateien zu einem Gazetteer zusammen. Üblicherweise sind die Listen in einzelne Kategorien (z.B. Name) und Unterkategorien (z.B. Vorname, Nachname, Titel, etc.) gegliedert.

Die Erstellung solcher Gazetteer Listen, welche üblicherweise domänen- und sprachspezifisch sind, erfolgt meist manuell von Experten. Diese Aufgabe ist sehr zeitintensiv und erfordert langfristige Anpassungen. Einer der größten Vorteile von manuell erstellten Gazetteer Listen ist, dass diese auf die exakten Gegebenheiten eines Korpus angepasst werden können. Auch lokale und weniger bekannte Personen und Organisationen können in den Listen erfasst werden, was ein weiterer Vorteil ist. Maynard et al stellen in [12] eine Möglichkeit vor, sprachunspezifische Gazetteer Listen automatisiert basierend auf der Analyse linguistischer Korpora zu erstellen. Die vorgestellte Methode ist zwar sehr nützlich, jedoch lässt die Qualität der Ergebnisse zu wünschen übrig [12].

DBpedia Gazetteer Listen

DBpedia Ressourcen als Gazetteer zu verwenden erlaubt es hingegen, hoch qualitative Listen zu erstellen, welche durch die kollektive Arbeit der Wikipedia Community entstanden und zahlreichen Qualitätskontrollen unterlaufen sind.

Zahlreiche Arbeiten ([16–18]) belegen, dass eine kollaborativ erstellte Wissensressource wie Wikipedia eine wertvolle lexikalisch semantische Datenbank mit einem hohen Potential für den Einsatz in Computerlinguistik Applikationen darstellen.

Die englische Version der DBpedia Wissensdatenbank umfasst zum Zeitpunkt des Verfassens dieser Arbeit 3,77 Millionen Datensätze.[39] Davon sind 2,35 Millionen in einer konsistenten Ontologie klassifiziert. Jeder dieser Datensätze wird als „Thing“ beschrieben und kann einer mit englischem Namen versehenen Kategorie zugeordnet werden. Die Kategorien unterscheiden sich allerdings von jenen, welche bei Wikipedia verwendet werden. Während Wikipedia Artikel einer Domäne zuordnet, werden diese bei DBpedia eher einem Typ zugeordnet [2, S. 66].

- Wikipedia Toplevel Kategorien: General reference, Culture and the arts, Geography and places, Health and fitness, History and events,

[38] http://de.wikipedia.org/wiki/_text
[39] http://wiki.dbpedia.org/Datasets#h18-3

Mathematics and logic, Natural and physical sciences, People and self, Philosophy and thinking, Religion and belief systems, Society and social sciences, Technology and applied sciences.

- DBpedeia Toplevel Kategorien: Activity, AnatomicalStructure, Award, Beverage, ChemicalCompound, Currency, Device, Disease, Drug, EthnicGroup, Event, Infrastructure, Language, MeanOfTransportation, MusicGenre, Organisation, Person, Place, Planet, Protein, Species, Website, Work.

Diese Kategorisierung kommt der Verwendung in der Entitätenextraktion sehr entgegen, da das Ziel der Entitätenextraktion unter anderem deren Kategorisierung in ein Set von vorgegebenen Typen ist: Person, Organisation, Beruf, Ort, Datum-, Währungs und Zeitangaben. Leider sind diese jedoch größtenteils nur für Things der englischen DBpedia verfügbar. Eine Nachverfolgung der Links zu den mehrsprachigen Versionen eines englischsprachigen Artikels erlaubt es allerdings, die benötigten Kategorien auf die deutschsprachigen Versionen zu übertragen.

Obwohl es sich bei dem erstellten Gazetteer um eine linguistische Ressource für Einsatzzwecke in deutschsprachigen Applikationen handeln soll, wurden sowohl die Things der englischsprachigen, als auch jene der deutschsprachigen DBpedia verwendet. Der Grund dafür ist, dass die englische Version umfassendere Artikel zu Personen, Orten und Organisationen enthält. Somit wird das Maximum an möglichen Entitäten erkannt und gleichzeitig werden spezifische deutschsprachige Entitäten berücksichtigt. Das Ergebnis ist ein umfassender Gazetteer, welcher im Rahmen der Computerlinguistik auf deutschsprachige Korpora zur Entitätenextraktion eingesetzt werden kann.

3.3.3 Gazetteer Regeln

Die Gazetteer Regeln und Grammatiken bilden die eigentliche Logik und Intelligenz bei der Erkennung von Entitäten ab. Nachfolgend werden die wichtigsten Schritte, welche mit automatenbasierten Grammatiken abgebildet wurden, erläutert.

Label

In die Entitätenlisten werden hauptsächlich die Titles der Wikipedia Artikel eingefügt. Diese kurzen und bündigen Titel erlauben eine einwandfreie Zuordnung bei einer Analyse, welche die Groß- und Kleinschreibung nicht berücksichtigt. Etwaige Zusätze in Klammern werden für die Gazetteer Liste entfernt. Zum Beispiel die Stadt „Obama“ in der Präfektur Fukui, Japan [[Obama (Fukui)]] wird zu [[Obama]] in der Gazetteer Liste.

Tabelle 3.5: Klassifikationsschema der DBpedia Kategorien zur Umlegung den Entitätentyp „Location“.

Location
http://dbpedia.org/ontology/Settlement
http://dbpedia.org/ontology/PopulatedPlace
http://dbpedia.org/ontology/Place
http://dbpedia.org/ontology/Country
http://dbpedia.org/ontology/Lake
http://dbpedia.org/ontology/BodyOfWater
http://dbpedia.org/ontology/NaturalPlace
http://dbpedia.org/ontology/River
http://dbpedia.org/ontology/Stream
http://dbpedia.org/ontology/MountainRange
http://dbpedia.org/ontology/Island
http://dbpedia.org/ontology/City
http://dbpedia.org/ontology/Mountain
http://dbpedia.org/ontology/AdministrativeRegion
http://dbpedia.org/ontology/ProtectedArea
http://dbpedia.org/ontology/Atoll
http://schema.org/Place
http://schema.org/Country
http://schema.org/BodyOfWater
http://schema.org/LakeBodyOfWater
http://schema.org/RiverBodyOfWater
http://schema.org/City
http://schema.org/Mountain
http://schema.org/Canal

Kategorisierung

Um die DBpedia Kategorien auf die benötigen Typen von Entitäten umzulegen wurde ein Klassifikationsschema eingeführt, welches DBpedia Kategorien eindeutig in Entitätentypen umlegt. Für sämtliche Kategorien, welche nicht in das Schema der Entitätentypen passen, wurde eine weitere Kategorie vom Typ Thing eingeführt. Eine zusätzliche Optimierung wäre es für diesen Zweck weitere Typen einzuführen. Siehe Tabellen 3.5, 3.6 und 3.7.

Entitätenerkennung

Die Entitätenerkennung beginnt mit dem *Pre-Processing* und der flachen Satzverarbeitung. Dabei wird der Text in einfache sprachlich relevante Einheiten zerlegt, welche anschließend als Token Annotationen abgespeichert werden. Im nächsten Schritt, dem Part-of-speech Tagging werden die Token mit Zusatzinformationen über die Wortart angereichert. Diese Information ist für die Entitätenerkennung äußerst wichtig, da nur gewisse Formen von Nomina eine Entität abbilden können. In der Entitätengrammatik wird somit überprüft, ob es sich bei dem Token um eine Form von einem

Tabelle 3.6: Klassifikationsschema der DBpedia Kategorien zur Umlegung auf den Entitätentyp „Person“.

Person
http://dbpedia.org/ontology/Person
http://schema.org/Person
http://dbpedia.org/ontology/Astronaut
http://dbpedia.org/ontology/PokerPlayer
http://dbpedia.org/ontology/RugbyPlayer
http://dbpedia.org/ontology/SoccerPlayer
http://dbpedia.org/ontology/Athlete
http://dbpedia.org/ontology/GolfPlayer
http://dbpedia.org/ontology/FigureSkater
http://dbpedia.org/ontology/Wrestler
http://dbpedia.org/ontology/IceHockeyPlayer
http://dbpedia.org/ontology/BadmintonPlayer
http://dbpedia.org/ontology/TennisPlayer
http://dbpedia.org/ontology/Cyclist
http://dbpedia.org/ontology/BaseballPlayer
http://dbpedia.org/ontology/BasketballPlayer
http://xmlns.com/foaf/0.1/Person
http://dbpedia.org/ontology/FormulaOneRacer
http://dbpedia.org/ontology/AmericanFootballPlayer
http://dbpedia.org/ontology/GridironFootballPlayer

Nomen handelt (z.B. categoryTokenFeature.equals("NN") || categoryTokenFeature.equals("NE")). Alle anderen Satzteile werden von der Analyse vorab ausgeschlossen, was neben einer Qualitätssteigerung auch zu einer Performanzoptimierung führt. Wenn diese Überprüfung nicht durchgeführt wird, könnte zum Beispiel das Determinativ „Die“ fehlerhaft als Stadt im französischen Drôme (http://en.wikipedia.org/wiki/Die,_Drôme) erkannt werden.

Disambiguierung und Tippfehler

Um Ambiguitäten, Rechtschreibfehler und Tippfehler aufzulösen, werden sowohl Informationen von den Disambiguierungs-Seiten als auch die Wikipedia Weiterleitungen verwendet. Eine weitere wertvolle Quelle um Synonyme und unterschiedliche Schreibweisen zu erkennen sind die Informationen der Wikipedia Hyperlinks. Durch die vielfältige Verwendung in unterschiedlichsten Artikeln entsteht eine Vielzahl an unterschiedlichen Schreibweisen und Namensgebungen für einen Artikel. Aus diesem Grund ist die Information aus diesen Quellen meist noch vollständiger als jene der Disambiguierungs-Seiten und Weiterleitungen. Für ein optimales Ergebnis werden die Informationen aus allen Quellen zusammengefügt.

Während zur Korrektur von Rechtschreib- oder Tippfehlern einfach die Informationen der Wikipedia Weiterleitungen verwendet werden, ist die Auflösung von Ambiguitäten komplexer. Die Informationen aus Wikipedia lie-

Tabelle 3.7: Klassifikationsschema der DBpedia Kategorien zur Umlegung auf den Entitätentyp „Organization“.

Organization
http://dbpedia.org/ontology/Company
http://dbpedia.org/ontology/Organisation
http://dbpedia.org/ontology/Non-ProfitOrganisation
http://dbpedia.org/ontology/PoliticalParty
http://dbpedia.org/ontology/University
http://dbpedia.org/ontology/EducationalInstitution
http://dbpedia.org/ontology/TelevisionStation
http://dbpedia.org/ontology/Broadcaster
http://dbpedia.org/ontology/Library
http://dbpedia.org/ontology/Building
http://dbpedia.org/ontology/ArchitecturalStructure
http://dbpedia.org/ontology/Airline
http://dbpedia.org/ontology/SoccerClub
http://dbpedia.org/ontology/SportsTeam
http://dbpedia.org/ontology/School
http://dbpedia.org/ontology/Hospital
http://dbpedia.org/ontology/Work
http://schema.org/Organization
http://schema.org/EducationalOrganization
http://schema.org/CollegeOrUniversity
http://schema.org/TelevisionStation
http://schema.org/Library
http://schema.org/School
http://schema.org/SportsTeam
http://schema.org/Hospital

fern dabei zwar vollständige und qualitativ hochwertige Listen an potentiellen Bedeutungen für einen Begriff, die Auswahl, welche der Bedeutungen im Einzelfall zutrifft, muss jedoch durch eine Heuristik erfolgen.

3.4 Relevanzbewertung

Nach der erfolgreichen Extraktion von Entitäten im Fließtext soll deren Relevanz berechnet werden. Dabei werden grundlegend zwei unterschiedliche Arten von Relevanz eingeführt: Die Globalrelevanz, welche eine Entität unabhängig von dem umgebenden Kontext bewertet und die Kontextrelevanz, welche die Relevanz einer Entität in Hinsicht auf das aktuelle Dokument berechnet.

3.4.1 Herangehensweise

Im ersten Schritt wird die Globalrelevanz berechnet. Um drastische Unterschiede zwischen einzelnen Entitäten im Dokument auszugleichen, erfolgt die Berechnung allerdings zum einen immer in Relation zu der relevantesten En-

tität im Dokument und zum anderen wird ein natürlicher Logarithmus auf die Ergebnisse angewendet um die Unterschiede zu normalisieren.

3.4.2 Verwendete Datenquellen

Zur Berechnung der Relevanz werden sowohl semantische Datenquellen wie etwa die Anzahl der eingehenden Links zu einem DBpedia Artikel, als auch Daten aus dem Social Web wie die Anzahl der Likes einer Facebook Page verwendet. Zusätzlich spielt auch die Google AdWords API eine entscheidende Rolle vor allem bei der Bewertung der Kontextrelevanz.

Globalrelevanzbewertung

Die Globalrelevanz einer Entität setzt sich aus drei fundamentalen Bestandteilen zusammen: Die Anzahl der eingehenden Links zu einer DBpedia Ressource (falls es zur Entität einen DBpedia Eintrag gibt und sie aufgrund von dieser erkannt wurde), die Anzahl der „Gefällt mir“ und „Sprechen darüber“ der ersten Facebook Page, welche bei der Suche nach dem Entitätenbegriff zurückgegeben wird, sowie grundlegende Daten von der Google AdWords API wie die globalen monatlichen Suchanfragen, die zu erwartenden Klicks und die Kosten pro Klick (CPC). Für jede einzelne Komponente wird basierend auf dem höchsten Wert im Dokument eine Relevanz zwischen null und eins errechnet. Sei $R_W(t)$ die Relevanz einer Entität t basierend auf den Daten von Wikipedia, berechnet sich die Relevanz aus der Anzahl der eingehenden Wikipedia Links $n(t_L)$ dividiert durch die höchste Anzahl der Links einer Entität im Dokument $n(t_L)_{max}$. Um die Unterschiede zwischen den einzelnen Ergebnissen auszugleichen wird ein natürlicher Logarithmus auf das Ergebnis angewendet. Um nur positive Ergebnisse zu erhalten wird 1,1 zu jedem Ergebnis addiert. Siehe Gl. 3.2, 3.3 und 3.4.

$$R_{Wikipedia}(t) = \log(\frac{n(t_{Links})}{n(t_{Links})_{max}} + 1.1, e) \tag{3.2}$$

$$R_{Facebook}(t) = \log(\frac{\frac{n(t_{Likes})}{n(t_{Likes})_{max}} + \frac{n(t_{Talking})}{n(t_{Talking})_{max}}}{2} + 1.1, e) \tag{3.3}$$

$$R_{AdWords}(t) = \log(\frac{\frac{n(t_{Clicks})}{n(t_{Clicks})_{max}} + \frac{n(t_{CPC})}{n(t_{CPC})_{max}} + \frac{n(t_{Searches})}{n(t_{Searches})_{max}}}{2} + 1.1, e) \tag{3.4}$$

Die Berechnung der einzelnen Relevanzen erfolgt immer nach dem gleichen Schema, während sich die Wikipedia Relevanz nur aus einem Parameter (der Anzahl der eingehenden Links) zusammensetzt, werden bei Facebook zwei Parameter und bei Google AdWords drei Parameter verwendet. Siehe Tabelle 3.8.

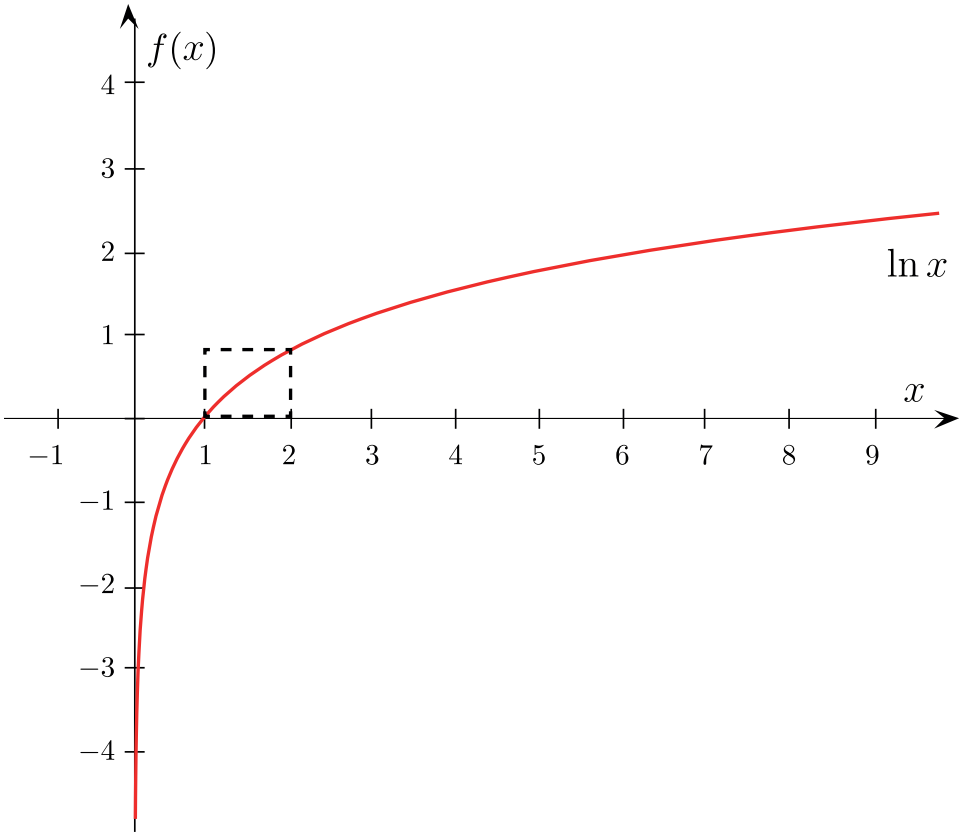

Abbildung 3.6: Umlegen der Relevanzergebnisse zwischen null und eins auf eine natürliche logarithmische Kurve im positiven Bereich.

Tabelle 3.8: Parameter zur Berechnung der einzelnen Relevanzen.

Wikipedia	Anzahl der eingehenden Links
Google AdWords API	Monatliche globale Suchanfragen
	Kosten pro Klick (CPC)
	Durchschnittliche Klicks pro Monat
Facebook	Anzahl der Likes
	Anzahl der Sprechen Darüber
Globale Relevanz	**Mittel der einzelnen Relevanzen**

Die Gesamt-Globalrelevanz R_{Global} ergibt sich durch das Addieren der einzelnen Relevanzen ($R_{Facebook} + R_{Wikipedia} + R_{AdWords}$) und das Dividieren durch die Anzahl der Einzelrelevanzen ($n(R)$) und stellt somit das Mittel dar. Eine Gewichtung der einzelnen Quellen wäre denkbar, wurde in diesem Fall aber nicht durchgeführt. Siehe Gl. 3.5.

$$R_{Global}(t) = \frac{R_{Wikipedia}(t) + R_{Facebook}(t) + R_{AdWords}(t)}{n(R)} \tag{3.5}$$

Kontextrelevanz

Die Kontextrelevanz versucht die Relevanz einer Entität im Bezug auf das die Entität beinhaltende Dokument festzustellen. Das Ziel ist es, zum einen diese Daten für weiterführende interne und externe computerlinguistische Anwendung zu verwenden und andererseits dem Leser zu vermitteln, wie relevant bestimmte Entitäten im Bezug auf den aktuellen Kontext sind. Wenn sich ein Artikel zum Beispiel hauptsächlich mit dem Thema Semantic Web befasst, soll dem Leser sofort ohne bestehendes Vorwissen ersichtlich sein, welche Personen, Organisationen und Orte in diesem Zusammenhang wichtig sind. Dieser Ansatz findet vor allem im Bereich des Wissensmanagements Verwendung.

Zur Berechnung der Kontextrelevanz werden ausschließlich Daten von der Google AdWords API verwendet. Spezifischer wird das *TargetingIdeaSelector* Service verwendet um eine Liste von verwandten Suchbegriffen für eine Entität zu erhalten. Diese Liste wird anschließend mit einer Liste von im Dokument vorkommenden Entitäten verglichen. Je mehr Überschneidungen es zwischen der Liste an verwandten Suchbegriffen einer Entität mit den Entitäten im Dokument gibt, desto höher ist der errechnete Wert für die Kontextrelevanz. Die verwandten Suchbegriffe werden von Google durch die durch User eingegeben Suchphrasen erzeugt und nach Popularität sortiert. Die Anzahl der Resultate variiert dabei ebenfalls nach Popularität des Suchbegriffs.

Sei $t_{RelatedKeywords}$ eine Liste von verwandten Suchbegriffen einer Entität t und $d_{Entities}$ die vollständige Liste an Entitäten im aktuellen Dokument d, so bildet der Durchschnitt $\cap$ dieser beiden Mengen die Anzahl der Überscheidungen $t_{Kontext}$ im Dokument zwischen verwandten Suchbegriffen einer Entität t und den im Dokument vorkommenden Entitäten. Siehe Gl. 3.6. Eine Visualisierung diese Sachverhalts ist in Abbildung 3.7 zu finden.

$$t_{Context} = t_{RelatedKeywords} \cap d_{Enities} \tag{3.6}$$

Die absolute Anzahl an Überschneidungen pro Entität $n(t_{Context})$ wird anschließend zur Berechnung der Kontextrelevanz zwischen null und eins verwendet. Die Berechnung geht wie bei der globalen Relevanz wiederum von der höchsten Anzahl an Überschneidungen im Dokument $n(t_{Context})_{max}$ aus. Zu große Unterschiede werden durch einen natürlichen Logarithmus ausgeglichen. Siehe Gl. 3.7.

$$R_{Context}(t) = \log(\frac{n(t_{Context})}{n(t_{Context})_{max}} + 1.1, e) \tag{3.7}$$

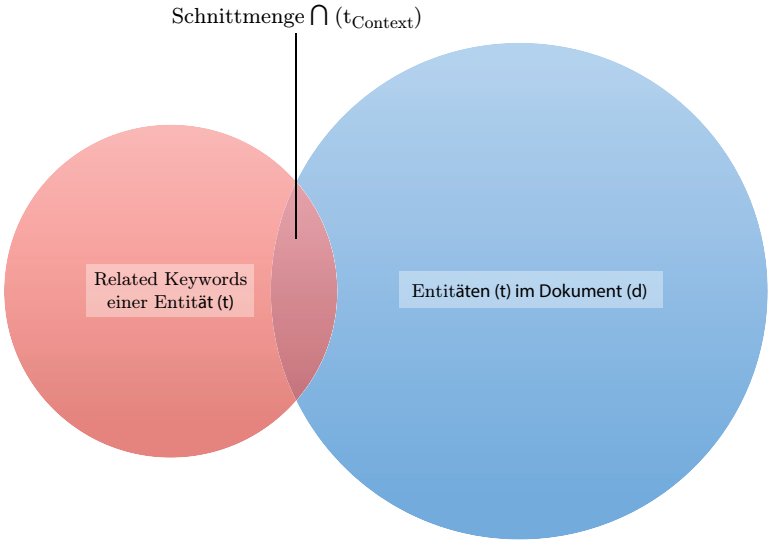

Abbildung 3.7: Visualisierung der absoluten Schnittmenge $t_{Context}$ aus den Mengen der Related Keywords einer Entität ($n(t_{RelatedKeywords})$ und der Entitäten im Dokument ($d_{Enities}$).

Kapitel 4

Prototypen Implementierung in GATE

Im letzten Kapitel wurden die Ansätze dieser Arbeit zur *Entitätenextraktion* und *Relevanzbewertung* mit Hilfe von externen Datenquellen auf einer theoretischen Ebene betrachtet. In diesem Kapitel soll nun das im Rahmen dieser Masterarbeit entstandene Masterprojekt vorgestellt werden. Zu Beginn wird ein kurzer Einblick in das zu Grunde liegende Text-Analyse Framework *GATE* (Abschnitt 4.1) und dessen auf Regular Expressions basierende Sprache *JAPE* (Abschnitt 4.2) gegeben. Anschließend wird die *Processing Pipeline* zur Text-Analyse vorgestellt und die wichtigsten Komponenten näher erläutert. Abschließend wird die auf Basis des *Spring Frameworks* implementierte *REST Schnittstelle* vorgestellt, welche die Komponenten der *Processing Pipeline* per *Webservice* verfügbar macht.

4.1 GATE (General Architecture for Text Engineering)

GATE ist eine auf JAVA basierende Infrastruktur zum Erstellen und Entwickeln von Software zur Verarbeitung von *natürlicher Sprache*. Die ersten Grundsteine zur Entwicklung von *GATE* wurde 1995 an der *University of Sheffield* in Großbritannien gelegt. Seither ist es nun schon fast 15 Jahre aktiv im Einsatz von zahlreichen internationalen computerbasierten Applikationen zur Verarbeitung von natürlicher Sprache [5].

GATE ist unter der *GNU Lesser General Public License (LGPL)*[1] als freie Software für Windows, OSX und Linux sowie als Open Source Projekt verfügbar. Die *LGPL* License ermöglicht sowohl das freie Nutzen der Software zu einem beliebigen Zweck als auch das Vervielfältigen und Weitergeben an Dritte. Der frei verfügbare Source Code kann des Weiteren beliebig verändert

[1] http://www.gnu.org/licenses/lgpl.html

werden, die einzige Prämisse ist, dass die veränderte Version ebenfalls unter der *LGPL*[2] (oder wahlweise der GPL) lizenziert werden muss [25, S.5].

Die Anzahl an verfügbaren *GATE* Ressourcen ist während der letzten Jahre ständig gestiegen. Die GATE Architektur beinhaltet heute folgende Komponenten[5, S.5-6]:

- eine *IDE*, **GATE Developer**: Eine integrierte Entwicklungsumgebung für Sprachverarbeitungsprozesse mit integrierten Komponenten zur Informationsextraktion sowie vielen weiteren Plugins. Siehe Abbildung 4.1.
- ein *Java Framework*, **GATE Embedded**: Eine Java Objektbibliothek, welche für die Einbindung von sämtlichen Services des *GATE Developers* in andere Applikationen vorgesehen ist.
- eine *cloud computing* Lösung **GATE Cloud** zum Verarbeiten von großen Text-Mengen in der Cloud.
- eine Web-Applikation **GATE Teamware** zum kollaborativen Annotieren von Texten online.
- ein *Such-Repositorium* **GATE Mímir**, welches zur Indizierung und zur Suche über Korpora, Texte, Annotationen und semantische Schemen (Ontologien) verwendet werden kann.

Die GATE Architektur besteht aus einzelnen Komponenten, so genannten *Ressourcen* zur Verarbeitung von natürlicher Sprache. Technisch gesehen sind GATE Komponenten spezielle Typen von Java Beans. Es gibt dabei folgende Spezifikationen [5, S.9-10]:

- **Language Resources (LRs)** repräsentieren Lexika, Korpora oder Ontologien.
- **Processing Resources (PRs)** repräsentieren Entitäten, welche vorrangig algorithmischer Natur sind, wie etwa Parser oder Generatoren.
- **Visual Resources (VRs)** repräsentieren visuelle Komponenten, welche in der GUI des GATE Developers vorhanden sind.

Zusammengefasst werden die in GATE bereits integrierten Komponenten als *CREOLE* bezeichnet, was in Englisch für „a Collection of REusable Objects for Language Engineering“ steht. Sämtliche Komponenten sind als Java Archive (JAR) gepackt und enthalten eine XML Konfigurationsdatei. Die Komponenten werden dabei mit Hilfe der GATE Embedded API (4.1.1) als Java Beans erstellt. GATE Developer wird dabei zur Visualisierung der Ergebnisse und zum Debugging verwendet.

GATE wird des Weiteren mit ANNIE (A Nearly-New Information Extraction System) ausgeliefert, welches ein Set an grundlegenden Komponenten darstellt, welches unter anderem einen Tokenizer, einen statischen Gazetteer, einen Sentence-Splitter, einen Wortart Tagger sowie einen Entitäten

[2] http://www.gnu.org/licenses/gpl-3.0

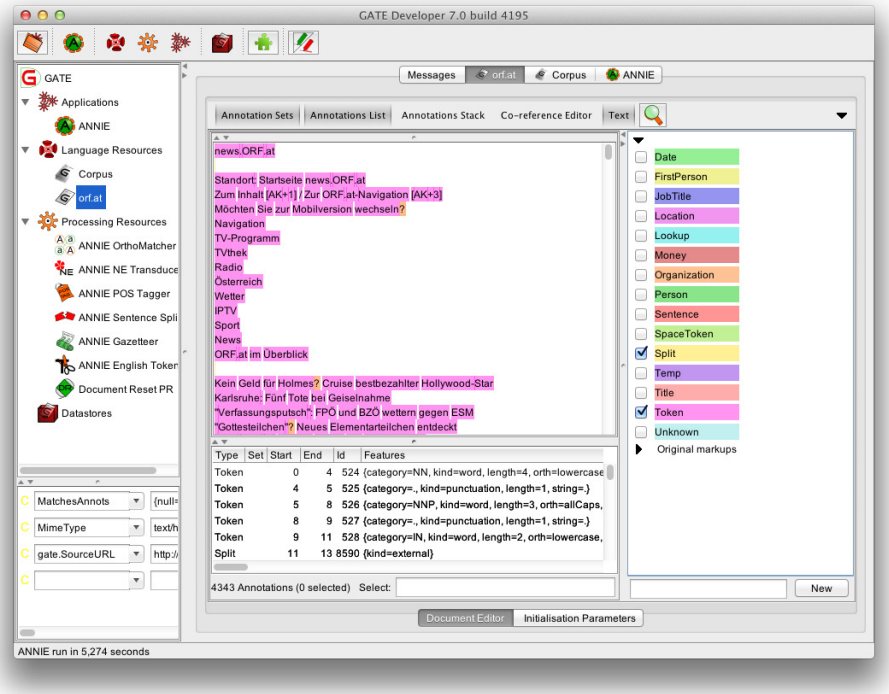

Abbildung 4.1: GATE Developer als visuelle Ressource zur Darstellung der Annotationen erstellt durch ANNIE.

Transduktor beinhaltet. ANNIE kann somit als Basis verwendet werden um grundlegende Informationsextraktionsfunktionalitäten abzubilden.

GATE inkludiert eine Vielzahl an üblichen Language Ressources (LRs), welche unterschiedliche Formate wie XML, RTF, E-Mail, HTML, SGML und Plaintext unterstützen. Somit können auch sämtliche auf XML basierende Dokumente wie jene von Microsoft Word, Powerpoint oder Excel seit dem 2007 eingeführten Office Open XML Format[3] eingelesen werden.

4.1.1 GATE Embedded

GATE Embedded ist die mit GATE ausgelieferte objektorientierte Java API, welche sämtliche Funktionen des *GATE Developer* beinhaltet und somit das Erstellen von Ressources, das Anpassen von Views und das Einbinden der GATE Funktionalität in Drittanwendungen erlaubt. Abbildung 4.2 visualisiert die Architektur der API und die explizite Trennung von Ressourcentypen. So werden verschiedene *Language Ressources* im Document Format Layer importiert, um nachfolgend durch verschiedene *Processing Ressour-*

[3] http://de.wikipedia.org/wiki/Office_Open_XML

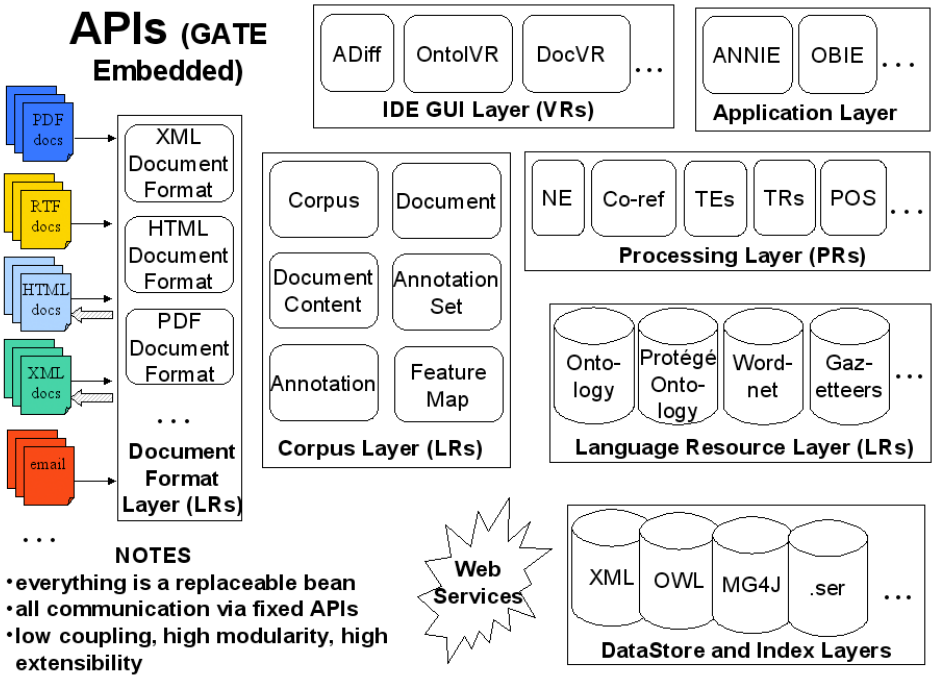

Abbildung 4.2: Die Architektur der GATE Embedded API visualisiert.

cen mit Zusatzinformationen in Form von Annotation Sets, Annotationen und Features angereichert zu werden. Einzelne Dokumente können des Weiteren auf der Ebene des Corpus Layers zu einem Korpus zusammengeführt werden. Der Language Ressource Layer bietet Language Ressources an, welche aber im Gegensatz zu den vorigen LRs nicht selbst annontiert werden, sondern nützliche Informationen zur Annotation bereitstellen. Dabei kann es sich zum Beispiel um eine Ontologie oder einen Gazetteer handeln. Der IDE GUI Layer stellt die visuellen Ressourcen des GATE Developer dar. Der DataStore and Index Layer ermöglicht das Speichern von in GATE erstellten oder annontierten Dokumenten und Korpora. Im Application Layer werden einzelne Processing Ressources zusammengefügt, welche nachfolgend in einer sogenannten Processing Pipeline auf einen Korpus von Dokumenten angewendet werden können.

4.2 JAPE: Regular Expressions over Annotations

JAPE steht für Java Annotation Patterns Engine und stellt einen auf Regular Expressions basierenden Transduktor zur Weiterverarbeitung von Annotationen zur Verfügung [5, S.183].

Eine JAPE Grammatik besteht aus verschiedenen Phasen, welche wie-

derum jeweils Muster- und Aktionsregeln beinhalten. Die linke Seite einer Grammatik (LHS engl. für left-hand-side) beschreibt die Regeln in Form von Annotationsmuster-Beschreibungen. Die rechte Seite (RHS engl. für right-hand-side) besteht aus Statements, welche die Annotationen manipulieren. So werden die Annotationen manipuliert, auf welche die Regeln im LHS Teil der Grammatik zutreffen [5, S.183]. Dieser Sachverhalt soll mit dem folgenden Codebeispiel näher erläutert werden [5, S.183]:

```
Phase: Jobtitle
Input: Lookup
Options: control = appelt debug = true

Rule : Jobtitle
(
{Lookup.majorType == jobtitle}
(
{Lookup.majorType == jobtitle}
)?
)
:jobtitle
-->
:jobtitle.JobTitle = {rule = 'Jobtitle'}
```

Der LHS Teil befindet sich vor dem „–>“ Zeichen und der RHS Teil direkt danach. Wie vorhin erwähnt, werden im LHS Teil die Regeln in Form von Mustern definiert. So wird in diesem Beispiel eine Regel mit dem Namen „Jobtitle“ definiert. Das Muster beschreibt eine Annotation vom Typ „Lookup“ welche ein Feature namens „majorType“ mit dem Inhalt „jobtitle“ beinhaltet, gefolgt von einer weiteren optionalen Annotation diese Typs. Sobald eine zu diesem Muster zutreffende Textsequenz gefunden wurde, wird der Sequenz ein Label zugeordnet (in diesem Beispiel „jobtitle“). Im RHS Teil (jener Teil nach dem „–>“) wird nun dieser Textteil über das zugeordnete Label angesprochen. In der darauffolgenden Aktion wird nun eine Annotation vom Typ „JobTitle“ erstellt und ein Annotations-Feature mit dem Namen „rule“ und dem Inhalt „JobTitle1“ erstellt. Am Anfang der Grammatik wird eine Phase mit dem Namen „Jobtitle“ definiert. Grammatiken können ineinander verschachtelt werden und so wird dieser Name unter anderem dafür verwendet, um den Namen für die generierte Java Klasse des RHS Teils zu bestimmen. Die Umwandlung in ein Java Objekt erlaubt des Weiteren die Verwendung von nativem Java Code im RHS Teil.

Eine weitere Besonderheit sind die Options der JAPE Grammatik. Hier kann zum Beispiel mit Hilfe des Attributes „control“ die Methode des Regel- und Annotationsvergleichs definiert werden. Hierfür gibt es wie in Tabelle 4.1 dargestellt fünf verschiedene Kontrollwerte.

Tabelle 4.1: Vergleich der unterschiedlichen control Optionswerte für den Regel- und Annotationsvergleich anhand der Beispielannotationen [aaa[bbb]] [ccc[ddd]] [5, S.203].

Option	Beschreibung
brill	Längste umschließende Annoation z.B.: [aaabbb][cccddd]
all	Alle Annotationen z.B.: [aaa[bbb]] [ccc[ddd]]
first	Erste gefundene Annotation
once	Maximal eine Regel und anschließendes Beenden der Phase
appelt	Nur eine einzelne Regel pro Annotation. Regel Prioritäten

4.3 Processing Pipeline

Die Processing Pipeline der SemantLink Applikation gliedert sich in vier Hauptphasen: Pre-Processing (4.3.1), Entitätenextraktion (4.3.2), Co-Reference und Relation Finding (4.3.3) und Relevanzbewertung (4.3.4). Jede Phase baut auf die vorhergehenden Phasen auf und besteht aus einzelnen Komponenten. In diesem Kapitel soll ein Einblick in den Aufbau der Pipeline und deren Zusammensetzung gegeben werden. Die Implementierungsdetails der generischen Komponenten wie dem Document Reset werden nur oberflächlich behandelt, während die applikationsspezifischen Komponenten wie Entitätenextraktion und Relevanzbewertung genauer beschrieben werden. Der vollständige Source Code der Applikation steht im Anhang der Arbeit sowohl als URL Ressource als auch auf einem physikalischen Datenträger zur Verfügung. Die Applikation wurde mit Hilfe der GATE Embedded API in Java modelliert. Die API Aufrufe sind dabei relativ einfach gehalten und initialisieren jeweils einzelne Processing Ressources (PRs), konfigurieren diese und fügen diese zur Processing Pipeline der Applikation hinzu. Die einzelnen PRs sind dabei spezielle Typen von Java Beans. Die einzelnen Schritte werden im folgenden näher erläutert. Eine Visualisierung der Pipeline ist in Abbildung 4.3 zu finden.

4.3.1 Pre-Processing

Die Pre-Processing Pipeline verwendet größtenteills fertige Ressourcen von *ANNIE*, welches zu Beginn dieses Kapitels bereits kurz vorgestellt wurde. Die einzelnen Ressourcen sind speziell auf die Anforderungen des Implementierungs-Prototypen angepasst und verwenden teilweise weitere externe Frameworks und Tools wie z.B. den TreeTagger.

Document Reset

Die *Document Reset* Ressource setzt die Input Dokumente in ihren Originalzustand zurück, in dem alle Annotation Sets und deren Inhalte gelöscht

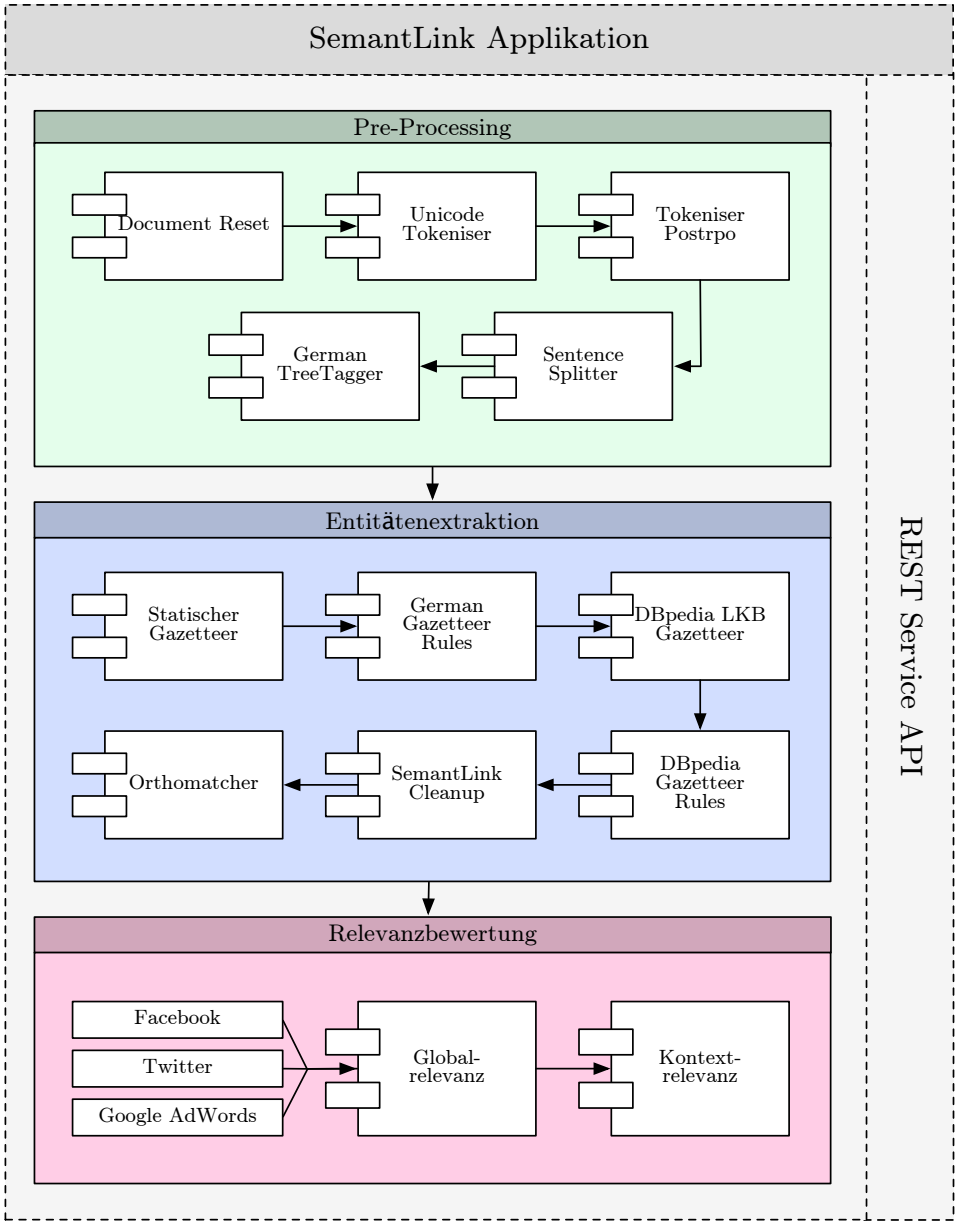

Abbildung 4.3: Processing Pipeline der SemantLink Applikation

werden. Davon ausgenommen ist das Annotations Set „Original Markups“, welches durch eine Format Analyse Strukturinformationen des Dokuments in Annotationen umwandelt. Somit werden Annotations-Duplikate und Inkonsistenzen bei mehrfachem Ausführen der Processing Pipeline auf das selbe Dokument verhindert [5, S.115].

Unicode Tokeniser

Der *Unicode Tokeniser* kümmert sich um die *flache Satzverarbeitung* und unterteilt den Text in einfache sprachlich relevante Einheiten wie z.B. Wörter, Nummern, Leer- oder Satzzeichen. Das Resultat dieser Ressource sind zwei Annotation Sets: Token und Space Token. Jede Annotation ist mit folgenden Attributen, so genannten Features angereichert: (Sub-)Typ (z.B. Word, Space), Länge, Groß- und Kleinschreibung. Die Tokenisierung bildet die Grundlage für alle weiteren Operationen und Annotationen, da sie die grundlegenden und kleinsten Elemente festlegt [5, S.117]:

Die Regeln des Unicode Tokenisers sind in *JAPE* modelliert und bestehen somit aus einem LHS Teil, welcher aus Regular Expressions besteht und einem RHS Teil, welcher die Annotationen und deren Features definiert:

```
{LHS} > {Annotation type};{attribute1}={value1};...;{attribute n}={value n}
```

Folgende Typen von Token und SpaceToken sind möglich:

Wort Ein Wort ist definiert als eine Gruppe von Groß- und Kleinbuchstaben sowie Bindestrichen. Jedes Wort hat ein Zusatzattribut mit dem Namen „orth“ welches mit folgenden Werten belegt werden kann:

- upperInitial - Wort beginnt mit einem Großbuchstaben, der Rest sind Kleinbuchstaben
- allCaps - Das Wort besteht nur aus Großbuchstaben
- lowerCase - Das Wort besteht nur aus Kleinbuchstaben
- mixedCaps - Jede andere Kombination an Groß- und Kleinbuchstaben, welche die obigen Kriterien nicht erfüllt.

Nummer Eine Nummer ist definiert als eine einzelne Ziffer oder als mehrere aufeinanderfolgende Ziffern mit Punkt- oder Kommatrennzeichen.

Symbol Es gibt zwei verschiedene Typen von Symbolen. Währungs-Symbole (z.B. $) und Symbole (z.B. &). Eine Zusammensetzung von mehreren aneinandergereihten Symbolen ist ebenfalls möglich.

Satzzeichen Folgende Satzzeichen sind definiert: „Start Punctuation“ (z.B. „) „End Punctuation“ (z.B. „!“) und „Other Punctuation“ (z.B. „:“). Jedes einzelne Satzzeichen ist ein eigener Token, eine Aneinanderreihung oder Kombination ist hier nicht möglich.

Leerzeichen Leerzeichen sind unterteilt in zwei Typen: „space“ und „control“, je nachdem ob es sich um reine Leerzeichen oder um bestimmte Kontrollleerzeichen handelt. Diese Annotationen werden unter dem Typ „SpaceToken“ zusammengefasst.

Tokeniser Postpro

Die Tokeniser Postpro (Nachbearbeitung) verwendet die *JAPE Transducer* Processing Ressource um JAPE Regeln auf die im vorigen Schritt erstellten Annotationen anzuwenden. Die Regeln sind dabei größtenteils auf das Zusammenführen von einzelnen Token zu sinngemäßen Einheiten ausgelegt. z.B. „40er“, sowie das Erkennen von Zeilenumbrüchen (z.B. \n) und das Anreichern von Annotationsfeatures mit Zusatzinformationen (z.B. Number Type).

Sentence Splitter

Die *Sentence Splitter* Ressource besteht aus einer Reihe von endliche Automaten, welche den Text in Satz-Annotationen gliedern. Um das Satzende korrekt zu erkennen wird eine statische Liste an Stoppwörtern verwendet. Es werden dabei Annotationen vom Typ „Sentence“ erstellt. Im Detail handelt es sich dabei um Annotationen vom Typ Sentence, welche den ganzen Satz umfassen und „Split“ Annotationen, welche die Leerzeichen zwischen den einzelnen Sätzen auszeichnen. Das Feature „kind“ beschreibt, ob es sich dabei um einen Satz innerhalb eines Absatzes (internal) oder am Ende eines Absatzes (external) handelt [5, S.121].

German TreeTagger

Das Tagger Plugin stellt einen generischen Wrapper zur Verfügung, welcher die Einbindung von diversen externen Taggern ermöglicht. Bekannte Tagger sind zum Beispiel externe Applikationen wie der *TreeTagger*[4] oder der *Stanford Tagger*[5]. Das Framework stellt dabei die Schnittstelle von GATE zu diesen Applikationen dar: Es liefert die benötigten Daten, welche bereits durch den Tokeniser und den Sentence Splitter vorliegen, an den Tagger weiter, welcher als Konsolenapplikation betrieben werden muss. Der Output des Taggers, welcher per „stdout“ Befehl erfolgen muss, wird durch das Framework wieder in die GATE Annotationen eingefügt. Die bestehenden Token Annotationen werden dabei mit den zusätzlichen Features „Category“ und „Lemma“ angereichert. Category beschreibt dabei die Wortart im vorkommenden Satz und Lemma beinhaltet die Grundform des Wortes (z.B. „Haus“ vom Wort „Häuser“) [5, S.442-443].

In der SemantLink Applikation wird der *TreeTagger* in seiner deutschen Ausführung verwendet. Neben Deutsch unterstützt der TreeTagger außerdem unter anderem noch Englisch, Französich, Italienisch, Spanisch und Russisch. Der TreeTagger wurde am Institut für Computerlinguitsik an der Unviversität von Stuttgart entwickelt.

[4] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/
[5] http://nlp.stanford.edu/software/tagger.shtml

4.3.2 Entitätenextraktion

Die Entitätenextraktion besteht aus einer Kette von Processing Ressourcen: Im ersten Schritt wird ein statischer ANNIE Gazetteer angewendet, welcher Strings mit Hilfe von statischen deutschsprachigen Listen erkennt. Die tatsächliche Logik der statischen Analyse ist nachfolgend in einen JAPE Transduktor ausgelagert. Dieser fügt mit Hilfe von in JAPE definierten Regeln die einzelnen erkannten Strings in logische Gruppen und Entitäten zusammen. Anschließend wird ein Large Knowledge Base (LKB) Gazetteer angewendet, welcher zum Zeitpunkt des Verfassen dieser Arbeit 3,77 Millionen[6] DBpedia Ressourcen beinhaltet. Die Inhalte werden dabei dynamisch von der mit semantischen Technologien zugänglichen Webressource *DBpedia* extrahiert, welche sämtliche Wikipedia Artikel in regelmäßigen Abständen abgreift und in eine Ontologie umwandelt. In einem letzten Schritt werden durch einen weiteren JAPE Transduktor die erkannten DBpedia Entitäten verfeinert und mit den statischen Entitäten zusammengefügt.

Statischer Gazetteer

Der *ANNIE Gazetteer* ist die übliche Methode um Entitäten mit Hilfe von GATE im Rahmen der Computerlinguistik zu extrahieren. Diese Processing Ressource erlaubt es, statische Listen von Entitäten im *Plaintext*[7] Format zu definieren und sie anhand von diesen in Texten wiederzuerkennen. Die jeweiligen Listen sind dabei in einzelne Dateien ausgelagert, welche pro Zeile eine Entität beinhalten und jeweils logische Gruppen repräsentieren z.B. Namen von Städten, Organisationen, Datumsangaben, etc. Die einzelnen Dateien werden schließlich in einer Index Datei (lists.def) zusammengefügt, welche der Processing Ressource als Referenz übergeben wird. Für jede Liste ist ein „Major“ Typ (z.B. Name) und optional ein „Minor“ Typ (z.B. Vorname, Nachname, Titel, etc.) spezifiziert. Diese Klassifizierung hilft bei der anschließenden Zusammenführung von erkannten Strings zu Entitäten und logischen Gruppen. Sämtliche erkannte Strings werden durch eine Verarbeitung von endlichen Automaten in Annotationen vom Typ „Lookup“ zusammengefügt. Die Features der Annotationen spezifizieren dabei den „Major“ und den „Minor“ Typ des Strings. Eine Visualisierung einer statische Gazetteer Ressource im GATE Developer ist in Abbildung 4.4 zu sehen.

Die tatsächliche Logik der statischen Analyse ist nachfolgend in einen JAPE Transduktor ausgelagert. Diverse Grammatik Regeln definieren so basierend auf den Lookup Annotationen und den zur Verfügung stehenden Informationen in den Features zusammengehörende Entitäten. So kann es sich bei einer Personenentität z.B. um den Zusammenschluss aus zwei Namen (Major Typ) Strings handeln: eine vom Minor Typ Vorname und eine vom

[6] http://dbpedia.org/About
[7] http://de.wikipedia.org/wiki/_text

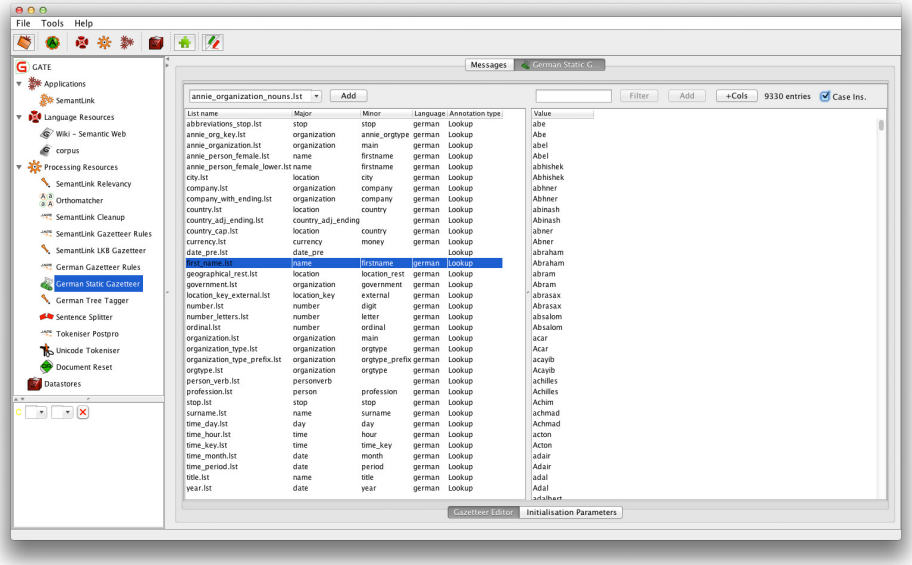

Abbildung 4.4: Statische Gazetteer Ressource im GATE Developer

Typ Nachname. Zusätzlich muss es sich im grammatikalischen Sinne um eine Nominal-Phrase (NP) handeln. Die letztere Information wird dabei aus den Features der Annotation vom Typ „Token“, welche im vorigen Schritt erstellt wurde, herangezogen. Dabei werden wiederum Features verwendet, die die Lage eines Strings im Text beschreiben: den Start und End Position Features. Auf diesem Weg werden Entitäten vom Typ Person, Organisation, Beruf sowie Ort, Datum-, Währungs und Zeitangaben erkannt. In der praktischen Anwendung wurden eine Vielzahl von weiteren komplexeren Szenarien implementiert, welche ebenfalls auf das Auftreten einer Entität eines bestimmten Typs hindeuten. Diese Regeln wurden mit einer bestimmten Namenskonvention ausgezeichnet: „Guess_...“. Ein solches Beispiel wird anhand einer JAPE Grammatik (4.1) näher erläutert: Die Regel „Guess_full_name-_by_profession“ identifiziert so wie im Beispiel-Code gezeigt wird eine Personenentität dadurch, dass es sich um ein bis zwei Strings beginnend mit einem Großbuchstaben direkt nach einer erkannten Entität vom Typ Beruf handelt. So könnte zum Beispiel „Obama“ als Person erkannt werden, wenn dieser in Kombination mit der Entität vom Typ Beruf „Präsident“ auftritt, obwohl der Name nicht in einer der statischen Gazetteer Listen vorkommt. Solche Regeln sind allerdings nicht in allen Fällen verlässlich und produzieren somit auch sogenannte „False-Positives“, also z.B. Personenentitäten, welche eigentlich gar keine sind. Eine Verfeinerung und Abstimmung dieser Regeln ist somit ein langfristiger Prozess, welcher längere Tests umfasst. Auch eine Spezialisierung der Regeln auf ein bestimmtes Themenumfeld ist denkbar

```
Rule: Guess_full_name_by_profession
Priority: 20
// e.g. Präsident Obama
({Jobtitle})+
(
  {Token.orth == upperInitial}
  ({Token.orth == upperInitial})?
):prof
-->
:prof.Person = {kind = "Person", rule = "Guess_full_name_by_profession"}
```

Programm 4.1: JAPE Grammatik zur Identifizierung von Personen Entitäten durch deren Vorkommen nach einer Entität vom Typ Beruf.

um die Qualität zu steigern. Mehr dazu findet sich im Kapitel 5.

DBpedia LKB Gazetteer

Der DBpedia Large Knowledge Base (LKB) Gazetteer stellt eine sehr wichtige Komponente des Prototypen gerade in Hinsicht auf diese Arbeit dar. Zusätzlich zu den im vorigen Schritt durch den statischen Gazetteer erkannten Entitäten ist es das Ziel, die Resultate der Entitätenextraktion durch das Hinzufügen von externen Datenquellen zu optimieren. Während die theoretische Herangehensweise bereits in Kapitel 3.3 beschrieben wurde, soll in diesem Kapitel die technische Implementierung nur auf einer oberflächlichen Basis beschrieben werden. Ziel soll es nicht sein die Implementierung im Detail zu beschreiben, hierfür ist der Source Code im Anhang dieser Arbeit gedacht.

Als Ausgangsbasis wurde die von *Ontotext*[8] entwickelte Processing Resource *The Large KB Gazetteer*[9] verwendet. Diese stellt Werkzeuge zur Verfügung um umfassende Vokabulare effizient als Gazetteer zu verwenden. Der Grund weshalb dieser Gazetteer als Grundlage ausgewählt wurde ist der, dass aus Performanzgründen ein Hash-Code indizierter Zwischenspeicher implementiert wurde, welcher als Adaption der Java HashSet Klasse genau an die Anforderungen eines Gazetteers angepasst wurde. Die fundamentalsten Änderungen sind, dass die Hashwerte extern kalkuliert werden und dass mehrere Objekte mit gleichen Hashwerten gespeichert und gruppiert werden können. Diese Gruppen sind als eigene interne Klassen definiert und können mit Hilfe eines bestimmten Hashwertes abgerufen werden.

Wie im Kapitel 3 ausführlich vermerkt wurde ist Performanz der kritischste Faktor für die Verwendung von externen Datenquellen in der Computerlinguistik. Gerade bei der Vielzahl von zu verarbeiteten Strings müssen

[8] http://www.ontotext.com/
[9] http://nmwiki.ontotext.com/lkb_Gazetteer/

HTTP Requests zu einem Minimum reduziert werden und möglichst viele Daten auf dem eigenen Server direkt zugänglich gemacht werden. Gleichzeitig müssen auch die servereigenen Ressourcen so weit wie möglich geschont werden um eine rasche Abwicklung zu gewährleisten.

Als Datenquelle wurde *DBpedia*[10] ausgwählt, welche die Informationen von *Wikipedia* in strukturierter und maschinenlesbarer Form auf Basis des *Semantic Web* frei zur Verfügung stellt. Als Standard für die Datenhaltung wird das *Resource Description Framework* (RDF) verwendet. Die Daten sind somit auch über einen öffentlich zugänglichen Endpoint mit der graphbasierten Abfragesprache *SPARQL* (SPARQL Protocol And RDF Query Lanuage)[11] abrufbar.

In einem ersten Schritt wurde versucht, diesen öffentlichen Enpoint[12] zu verwenden um auf die RDF Daten von DBpedia zuzugreifen. Nachdem bereits nach den ersten Tests bewusst wurde, dass die Abfragen auf diesem Weg viel zu lange dauern, wurde auch schnell die Restriktion bemerkt, dass über diesen öffentlichen Endpoint nur maximal 2000 Datensätze pro SPARQL Abfrage zurückgeliefert werden. Nachdem ein rekursiver Aufruf dieses Endpoints aus Performanzgründen ausgeschlossen werden musste, wurden die benötigten DBpedia Ressourcen auf dem eigenen Server repliziert. Dabei handelt es sich vorrangig um die Titel und Label der Datensätze. Dazu wurden zu Beginn die zum Download zur Verfügung stehenden Daten-Dumps im N-Triple Format verwendet. In einem weiteren Schritt wurde das *DBpedia Extraction Framework*[13] verwendet um aktuellere Datensätze in regelmäßigen Abständen zu erhalten.

Als Framework und als *Triplestore* wurde die auf Java basierende Open-Source Lösung *Sesame*[14] verwendet. Nach dem Import der benötigten N-Triple Sets: Titles[15], Ontology Infobox Types[16], Categories[17], Disambiguations[18]. Dabei wurden die Datensätze der deutschen und auch jene der englischen DBpedia verwendet, um sowohl im Deutschen relevante und sprachspezifische Entitäten zu finden als auch Entitäten, für welche es in der deutschen DBpedia keinen Eintrag gibt. Insgesamt umfasst die Datenbank zum Zeitpunkt des Verfassens dieser Arbeit 3,77 Millionen [19] DBpedia Ressourcen, während der statische Gazetteer nur 39.170 Entitäten beinhaltet. Eine Visualisierung des quantitativen Vergleichs der Datenmenge ist in Abbildung 4.5 zu sehen.

[10] http://dbpedia.org/About
[11] http://www.w3.org/TR/rdf-sparql-query/
[12] http://dbpedia.org/sparql
[13] http://mappings.dbpedia.org/index.php/Use_the_DBpedia_Extraction_Framework
[14] http://www.openrdf.org
[15] http://wiki.dbpedia.org/Downloads38#titles
[16] http://wiki.dbpedia.org/Downloads38#ontology-infobox-types
[17] http://wiki.dbpedia.org/Downloads38#categories-labels
[18] http://wiki.dbpedia.org/Downloads38#disambiguation-links
[19] http://dbpedia.org/About

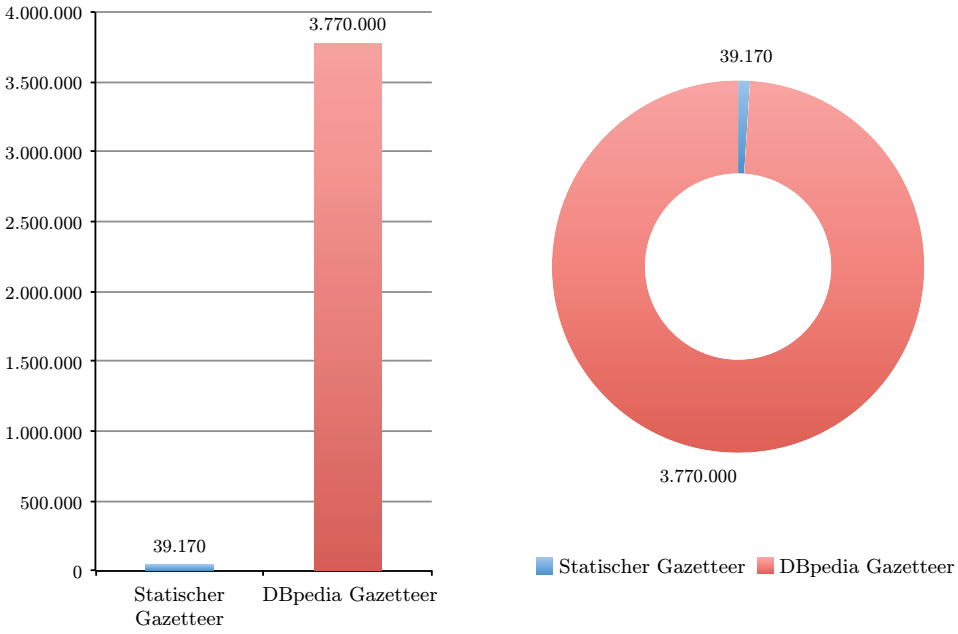

Abbildung 4.5: Statischer Gazetteer und DBpedia Gazetteer im quantitativen Vergleich

```
1  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns\#>
2  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema\#>
3
4  SELECT DISTINCT ?label ?s ?class WHERE{
5          ?s rdfs:label ?label .
6          ?s rdf:type ?class
7  }
```

Programm 4.2: SPARQL Abfrage zur Ausgabe säntlicher Ressourcen mit Label und Typ

Nachfolgend wird der am Server betriebene SPARQL Endpoint verwendet um die Abfragen möglichst performant und ohne Limitierungen auszuführen. Die SPARQL Query, wie in 4.2 illustriert, gibt sämtliche DBpedia Ressourcen, welche im Tripple Store hinterlegt sind zurück. Durch die „SELECT DISTINCT“ Abfrage werden Duplikate vermieden. Des Weiteren wird neben dem Label auch die zugehörige Klasse der Ressource zurückgegeben, um anschließend eine Zuweisung zu den unterschiedlichen Entitätentypen zu ermöglichen.

Die Resultate der Abfrage werden rekursiv in den vorhin erwähnten mit einem Hash Index versehen Zwischenspeicher abgelegt. Da die Berechnung des Cache je nach Umfang einiges an Zeit beansprucht (bei 3,77 Millionen Datensätzen bis zu 40 Minuten) und einen sehr hohen Arbeitsspeicheraufwand

verursacht, wird dieser nur bei fundamentalen Änderungen am Datenbestand neu erzeugt. Ansonsten verwendet die Processing Ressource den Hashed-Index Storage direkt als Gazetteer Liste, wobei durch dessen performanzorientierten Aufbau die Abfrage im Millisekunden Bereich durchgeführt werden kann. Die im Text wiedergefundenen Entitäten werden wie beim statischen Gazetteer als Annotationen mit zusätzlichen Features gespeichert. In diesem Fall werden die Annotationen als Typ „SemantLink_DBpedia“ abgespeichert und mit folgenden Zusatzinformationen hinterlegt: „Class“: Die DBpedia Klasse (z.B. http://dbpedia.org/ontology/Person), „Ressource“: Die DBpedia Ressource als URL (z.B. http://dbpedia.org/page/Barack_Obama, „Kind:“ Den Typ der DBpedia Ressource (z.B.: Person)

Der nächste Schritt ist ähnlich dem statischen Gazetteer in eine eigene Processing Ressource in der Pipeline ausgelagert. Ein JAPE Transduktor definiert auch hier wieder bestimmte Regeln und Grammatiken um die erkannten Entitäten in die richtigen Gruppen zusammenzufassen und Fehlannotationen zu eliminieren.

Ein letzter Transduktor mit dem Namen „SemantLink Cleanup“ hat die Aufgabe die Entitäten, welche mit dem statischen Gazetteer erkannt wurden, mit denen, die durch den DBpedia Gazetteer gewonnen wurden, zusammenzuführen. Bei einer Überschneidung wird der Konfidenzgrad der Validität der einzelnen Entität gestärkt. Dabei wird außerdem darauf geachtet, dass doppelte Annotationen zu einer einzigen zusammengeführt werden und sämtliche Zusatzinformationen in den Features der neuen Annotation vereinigt werden. Die Verwendung von beiden Gazetteer Varianten hat den Vorteil, dass sowohl alle in Wikipedia/DBpedia vorkommenden relevanten Entitäten als auch weniger relevante Personen und lokale Firmen, zu welchen es keinen Wikipedia/DBpedia Eintrag gibt, erkannt werden. Mehr zu den Ergebnissen und zu deren Evaluierung findet sich im Kapitel 5.

4.3.3 Koreferenz und Relationen

Zum Auffinden von Koreferenzen wurde das *Orthographic Coreference* (OrthoMatcher) Plugin verwendet und an die speziellen Anforderungen der eigens implementierten Entitätenextraktion angepasst. Die Aufgabe dieser Ressource ist es Verbindungen zwischen einzelnen Entitäten herzustellen um anschließend Koreferenzen zu finden und schlussendlich relevante Daten aus dem Text zu extrahieren.

Die Koreferenz und Relationsauflösung gliedert sich in einzelne Schritte: In der Vorverarbeitung (Preprocessing) werden die bestehenden „Sentence“ Annotationen verwendet um die einzelnen Sätze zu identifizieren. Für jeden Satz wird anschließend eine Liste an Annotationen, gruppiert nach Kategorie, erstellt. Ein besonderer Fokus wir in diesem Fall auf Personenentitäten gelegt. In einem weiteren Schritt wird versucht das Geschlecht der Personenentitäten zu identifizieren, dies ist allerdings nur bei vorhanden Vorna-

men oder durch Zusatzinformationen von DBpedia möglich. Beim nächsten Schritt handelt es sich um die Auflösung von Pronomen im Satz. Je nach Typ von Pronomen (z.B. ich, du, er, sie, mein(e), dein(e), sein(e), mich, dich, etc.) wird mit unterschiedlichen Algorithmen und Grammatiken versucht Relationen und somit Koreferenzen herzustellen. Die gewonnen Informationen lassen sich im GATE Developer mit Hilfe des „Co-reference Editors“ darstellen oder in einer Ontologie abspeichern.

4.3.4 Relevanzbewertung

Die *Relevanzbewertung* stellt einen weiteren wichtigen Bestandteil dieser Arbeit dar. Wie im vorigen Kapitel über die Entitätenextraktion soll auch hier wieder nur auf die speziellen Apsekte der praktischen Implementierung eingegangen werden. Das Ziel dieser Processing Resource ist das Bewerten der Relevanz der einzelnen Entitäten sowohl in globaler Hinsicht als auch in Hinsicht auf den Kontext des umgebenden Texts (Kontext Relevance). Dazu wurden verschiedene externe Datenquellen und APIs auf deren Brauchbarkeit evaluiert. Eines der wichtigsten Kriterien war dabei wiederum das Thema Performanz und die Möglichkeit Bulk-Requests abzusetzen um die Anzahl der HTTP-Requests zu minimieren. Als brauchbare Quellen haben sich dabei vor allem die *Google AdWords API*[20] und die *Facebook Open Graph API*[21] erwiesen. Beide diese Quellen erlauben es die Relevanz von Entitäten auf unterschiedliche Art und Weise zu beurteilen und ermöglichen das Senden von Bulk-Requests. Die *Twitter API*[22] wurde aufgrund der Limitationen[23] vorerst nicht in Betracht gezogen, da pro Stunde nur 350 Aufrufe möglich sind, was eine Verwendung in der Computerlinguistik momentan ausschließt.

Die Processing Ressource wurde basierend auf der *GATE Embedded API* als spezielle Form einer Java Bean implementiert. Die wichtigste architektonische Eigenschaft ist, dass die erkannten Entitäten ähnlich dem bei der Entitätenextraktion verwendeten Hash-Index in einer Hashmap gespeichert werden. Als Hashwert wird in diesem Fall der Name der Entität als String verwendet und als Wert wird ein Objekt vom Typ „SemantLinkRelevance“ hinterlegt. Dies ermöglicht eine Performanzoptimierung, einen vereinfachten Zugriff und das einmalige Bewerten von mehrfach vorkommenden Entitäten.

Ein „SemantLinkRelevance“ Objekt beinhaltet für jede Entität folgende Attribute: Google AdWords Cost Per Click (CPC), Google AdWords monatliche globale Suchanfragen, Google AdWords zu erwartende Klicks (Paid Clicks), Google AdWords Kontext Treffer (Context Matches), Anzahl der Facebook Likes. Für jedes Relevanzobjekt werden basierend auf diesen Daten

[20] https://developers.google.com/adwords/api/
[21] https://developers.facebook.com/docs/opengraph/
[22] https://dev.twitter.com/
[23] https://dev.twitter.com/docs/rate-limiting

```
1 public double calculateFacebookRelevanceRating(SemantLinkRelevanceObject
       _semantLinkRelevanceObject){
2   double facebookRating = 0.0;
3   int numberOfLikes = _semantLinkRelevanceObject.getFacebookLikes();
4   facebookRating = Math.log((double)numberOfLikes + 1.1) / Math.log((
      double)highestFacebookLikes + 1.1);
5   return facebookRating;
6 }
```

Programm 4.3: Berechnung der Facebook Relevanz

die folgenden Relevanzwerte berechnet: Facebook Relevanz, Google AdWords Relevanz, Gesamtrelevanz sowie Kontextrelevanz.

Facebook Relevanz

Die Facebook Relevanz berechnet sich für jede Entität aus der Anzahl der Facebook Likes für die durch die Facebook Open Graph Suche zurückgegebene Facebook Page[24]. Die Intelligenz, welche Page dabei die relevanteste für eine bestimmte Entität ist, wird somit bewusst an den Facebook Suchalgorithmus augelagert. Der Gedanke dabei ist, dass dem Facebook Suchalgorithmus eine Vielzahl an Daten zur Verfügung stehen, welche nicht über die API zugänglich sind und somit eine bessere Bewertung über die Relevanz der Suchergebnisse kaum möglich ist. Um die Relevanzwerte unter den einzelnen Entitäten im Dokument vergleichbar zu machen, wird jeder Wert basierend auf der Entität mit der höchsten Anzahl an Likes in diesem Dokument berechnet. Um die Unterschiede auszugleichen, wurde auf die Ergebnisse der Berechnung eine natürliche logarithmische Funktion angewandt. Das Resultat der Kalkulation ist ein Wert zwischen null und eins, welcher die Relevanz der jeweiligen Entität durch die Bewertung mit Hilfe von externen Datenquellen von Facebook beschreibt. Die Methode zur Berechnung der Facebook Relevanz ist in Programm 4.3 zu sehen.

Die Facebook API erlaubt es zudem gebündelte Bulk Requests abzusenden, sodass pro 300 Entitäten jeweils nur zwei HTTP API Abfragen durchgeführt werden müssen: Eine Abfrage zur Suche der passenden Facebook Page für jede Entität und eine um die Anzahl der Likes der jeweiligen Pages abzurufen. Dies führt zu einem immensen Performanz Vorteil gegenüber einer Implementierung, bei welcher für jede Entität zwei HTTP Requests abgesetzt werden müsste.

[24] https://www.facebook.com/about/pages

Google AdWords Relevanz

Die Google AdWords Relevanz berechnet sich aus den für den Entitäten-String durchgeführten Suchanfragen auf Google und den daraus berechneten Werten für das Google AdWords Werbemodell. Für jede Entität werden so die Kosten pro Klick (CPC), die monatlichen globalen Suchanfragen und die zu erwartenden Klicks für die Berechnung der Relevanz herangezogen. Dabei wird für jede einzelne Komponente ein Wert zwischen null und eins basierend auf dem höchsten Wert in seiner Kategorie im jeweiligen Dokument berechnet, ähnlich wie bei den Facebook Likes. Zur Normalisierung der Unterschiede wird wieder ein natürlicher Logarithmus verwendet. Die gesamte Google AdWords Relevanz berechnet sich als Mittel zwischen den drei zuvor kalkulierten Werten. Die benötigten Werte sind maschinenlesbar nur über die kostenpflichtige API zugänglich. Zur Einsicht kann jedoch auch das kostenlose Google AdWords Keyword Tool[25] verwendet werden. Siehe Abbildung 4.6.

Hinsichtlich Performanz erlaubt die Google AdWords API aufgrund ihrer ausschließlich kommerziellen Nutzung Bulk Requests bis zu einer Höhe von 2000 Abfragen pro API Aufruf. Jedoch müssen zwei unterschiedliche API Services verwendet werden: „TargetingIdeaService“ zur Abfrage der monatlichen globalen Suchanfragen und das „TrafficEstimatorService“ zur Abfrage der Kosten pro Klick (CPC) und der zu erwartenden Klicks (Paid Clicks). Somit sind für diese Relevanzberechnung 2 HTTP Requests pro 2000 Entitäten nötig. Die Methode zur Berechnung der Google AdWords Relevanz ist im Programm 4.4 ersichtlich.

Gesamt Relevanz

Die Gesamt Relevanz berechnet sich wiederum als Durchschnitt der Facebook und und der Google AdWords Relevanz. Der Ergebniswert spiegelt somit eine Art von Relevanz wider, welche zwar im Bezug zu anderen im Dokument vorkommenden Entitäten errechnet wurde, nicht aber unbedingt die Thematik und das inhaltliche Umfeld in die Kalkulation miteinbezieht. Zu diesem Zweck wurde versucht zusätzlich zu dieser Relevanz eine Art Kontextrelevanz zu errechnen, welche die Relevanz einer Entität im Bezug auf den Inhalt eines Dokumentes bestimmen kann.

Kontextrelevanz

Die Kontext Relevanz verwendet ebenfalls die *Google AdWords API* um die Relevanz einer Entität im Bezug auf den Inhalt des jeweiligen Dokumentes zu errechnen. Für diesen Zweck werden die mit der Entität verwandten Suchbegriffe (Related Keywords) betrachtet und mit allen anderen im Do-

[25] https://adwords.google.com/o/KeywordTool

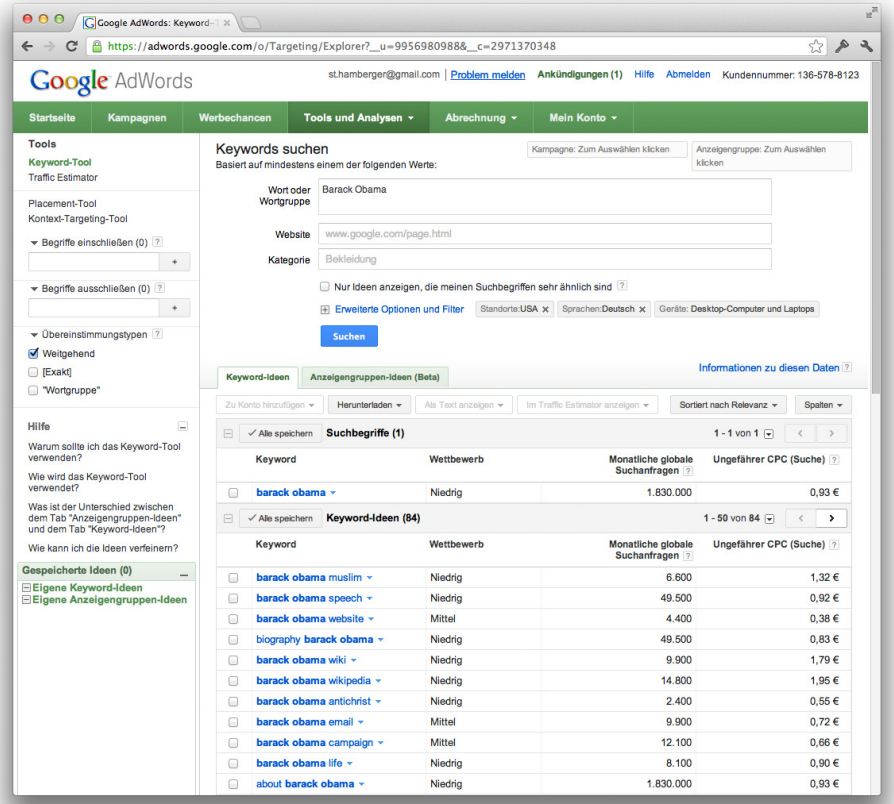

Abbildung 4.6: Google AdWords Keyword Tool

kument vorkommenden Entitäten verglichen. Je mehr Übereinstimmungen es zwischen der Liste an verwandten Suchbegriffen und den im Text vorkommenden übrigen Entitäten gibt, desto höher ist die Kontext Relevanz. Je nach Anzahl der Übereinstimmungen wird auch hier wieder ein normalisierter Wert basierend auf dem höchsten Übereinstimmungswert zwischen null und eins berechnet.

Die einzige Einschränkung besteht darin, dass es zur Zeit noch nicht möglich ist, Bulk Requests für das benötigte Google Ad-Words Service abzurufen. Somit muss pro Entität ein HTTP Request ausgeführt werden, wobei sich die Ausführungszeit je nach Anzahl der Entitäten erhöht. Ein Auszug des Quellcodes zur Abfrage der verwandten Suchbegriffe ist in Programm 4.5 zu sehen.

```
1 public double caculateAdWordsRelevanceRating(SemantLinkRelevanceObject
      _semantLinkRelevanceObject){
2   double adWordsCPCRating = 0.0;
3   double adWordsClicksRating = 0.0;
4   double adWordsRating = 0.0;
5   double adWordsMonthlySearchesRating = 0.0;
6   double adWordsCPC = _semantLinkRelevanceObject.getAdWordsCPC();
7   double paidClicks = _semantLinkRelevanceObject.getAdWordsPaidClicks();
8   double monthlySearches = _semantLinkRelevanceObject.
      getAdWordsMonthlySearches();
9
10  adWordsCPCRating =  Math.log(adWordsCPC + 1.1)/Math.log(
      highestAdWordsCPC + 1.1);
11  adWordsClicksRating = Math.log(paidClicks + 1.1)/Math.log(
      highestAdWordsPaidClicks + 1.1);
12  adWordsMonthlySearchesRating = Math.log(monthlySearches + 1.1)/Math.
      log(highestAdWordsMonthlySearches + 1.1);
13
14  adWordsRating = (adWordsCPCRating + adWordsClicksRating +
      adWordsMonthlySearchesRating) / 3;
15  return adWordsRating;
16 }
```

Programm 4.4: Berechnung der Google AdWords Relevanz

```
1 // Get related keywords.
2 TargetingIdeaPage relatedPage = targetingIdeaService.get(relatedSelector
      );
3
4 Map relatedKeywordMap = new HashMap();
5 int relatedKeywordMatchCount = 0;
6
7 // Display related keywords.
8 if (relatedPage.getEntries() != null && relatedPage.getEntries().length
      > 0) {
9   for (TargetingIdea targetingIdea : relatedPage.getEntries()) {
10    Map<AttributeType, Attribute> data = MapUtils.toMap(targetingIdea.
      getData());
11    keyword = (Keyword) ((CriterionAttribute) data.get(AttributeType.
      CRITERION)).getValue();
12    Long averageMonthlySearches = ((LongAttribute) data.get(
      AttributeType.AVERAGE_TARGETED_MONTHLY_SEARCHES)).getValue();
13 }
```

Programm 4.5: Berechnung der Kontext Relevanz mit Hilfe von Google AdWords Related Keywords

4.4 REST Service

Um die Ergebnisse des Implementierungs Prototypen per API zugänglich zu machen, wurde ein auf *Representational State Transfer* (REST)[26] basierendes Webservice als Java Spring Application entwickelt.

Als erste Ressource wurde „/process“ als ein „HttpRequestHandler“ Servlet definiert. Im Hintergrund befindet sich eine Java Bean vom Typ „Gate Pooled-Proxy“, welche das parallele Ausführen von mehreren „Handlern“ erlaubt. Jeder „Handler“ besteht aus einer vollwertigen GATE Applikation und beinhaltet somit die gesamte Processing Pipeline. Nach dem erfolgreichen Durchlauf werden die relevanten Entitäten als Objekte vom Typ „NamedEntity“ in einer FeatureMap abgelegt und anschließend im *JSON*[27] Format ausgegeben. Die REST Ressource kann somit per POST Request mit einem Plaintext, einer URL, einer XML basierten Datei oder ähnlichem angesprochen werden und liefert die extrahierten Entitäten im maschinenlesbaren JSON Format zurück. Die Entitätenextraktion kann somit für eine Vielzahl von Webapplikationen eingesetzt werden. Eine zweite Ressource ist unter dem Pfad „/relevance“ verfügbar. Zu diesem Endpoint können wahlweise und je nach Einsatzzweck vollständige Texte aber auch einzelne Entitäten gesendet werden. Der Output erfolgt wieder im JSON Format, die Entitäten sind hier allerdings zusätzlich mit den Globalrelevanz- und Kontextrelevanz-Attributen ausgestattet.

Als weitere Attribute werden im JSON die Position im Text und der Typ der Entität zurückgegeben. Dies ermöglicht unter anderem das Kennzeichnen oder Hinterlegen von Entitäten in einem Fließtext oder einer Website. Das Webservice wurde möglichst allgemein gehalten um eine Vielzahl von Applikationen zu ermöglichen. Je nach Einsatzzweck kann und soll die API jedoch um die gewünschten Anforderungen erweiter werden. Zu Demo Zwecken wurde ein kleines Web-Interface erstellt um die Möglichkeiten der API zu demonstrieren. Ein Screenshot ist in Abbildung 4.7 zu sehen. Der Source Code befindet sich ebenfalls im Anhang dieser Arbeit.

[26] http://de.wikipedia.org/wiki/Representational_State_Transfer
[27] http://en.wikipedia.org/wiki/JSON

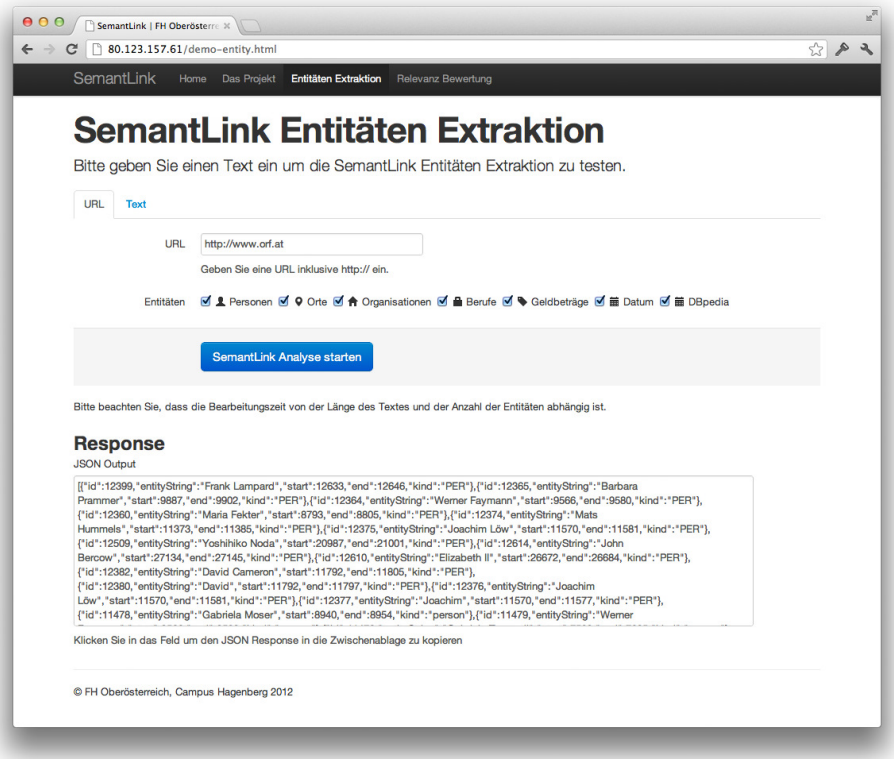

Abbildung 4.7: Screenshot der Demo Webapplikation mit REST Service Anbindung

Kapitel 5

Evaluierung

In diesem Kapitel sollen die Ergebnisse der beschriebenen und implementierten Methoden zur Entitätenextraktion und Relevanzbewertung evaluiert werden.

5.1 Entitätenextraktion

Da die Entitätenextraktion eine in der Computerlinguistik weit verbreitete und etablierte Anwendung ist, werden die Ergebnisse mit anderen bestehenden Algorithmen und Verfahren verglichen.

Um die Ergebnisse der Entitätenextraktion evaluieren zu können wird das in der Computerlinguistik und Informationsextraktion (IE) häufig verwendete *F-Maß*[1] (5.1.2) zur Bewertung der Performanz von Algorithmen und Methoden verwendet[11]. Um das F-Maß zu errechnen wird ein sogenannter Gold Standard (GS) benötigt, welcher als Korrektheitsstandard (5.1.1) dient. Nachfolgend werden die F-Maße der SemantLink Entitätenextraktion mit denen von anderen Entitätsextraktionsimplementierungen verglichen. Im Speziellen soll die von GATE ANNIE zur Verfügung gestellte statische Entitätenextraktion mit der in dieser Arbeit vorgestellten und in der SemantLink Applikation implementierten Methode der Entitätenextraktion mit Hilfe von DBpedia verglichen werden.

5.1.1 Korrektheitsstandard

Als Korrektheitsstandard dient der sogenannte Gold Standard, welcher händisch von Experten annotiert wurde und so ein nahezu perfektes Ergebnis liefert. Dieser Standard gilt als maximal zu erreichende Performanz und hat ein F-Maß von eins, wobei diese einen Wert von null (schlechteste) bis eins (bestes Ergebnis) darstellt. Für diesen Zweck gibt es in der englischen Sprache eine Vielzahl an linguistischen Korpora, welche bereits mit korrekt

[1] http://de.wikipedia.org/wiki/Beurteilung_eines_Klassifikators

annotierten Entitäten versehen sind. Ein bekanntes Beispiel dafür ist das MUC-7 Korpus[21], welches im Rahmen der letzten Message Understanding Conference (MUC)[2] im Jahre 1997 erstellt wurde. Dieses Korpus besteht aus englischsprachigen News Berichten zu den Themen Flugzeugabstürze und Raketenstarts. Da die entwickelten Methoden auf die Anwendung in der deutschen Sprache optimiert sind, wird allerdings ein deutschsprachiges Korpus benötigt. Zudem ist die im Rahmen des SemantLink Projekts entwickelte Applikation speziell auf eine Anwendung in kleinen und mittleren Unternehmen (KMUs)[3] in Österreich ausgelegt. Aus diesem Grund wurde ein passendes Korpus zusammengestellt und händisch annotiert.

Dokumente

Die gesammelten Dokumente wurden nach ihrer Herkunft und Zusammengehörigkeit auf fünf verschiedene Testkorpora eingeteilt. Zum einen wurden öffentlich verfügbare Dokumente von Firmenwebsites aus der Umgebung wie Pressemeldungen oder Website Artikel gesammelt. Zum anderen wurden auch interne Unternehmensdokumente wie E-Mails oder Projektdokumentationen der FH-Oberösterreich, Campus Hagenberg und der F&E in das Korpus mitaufgenommen.

Entitäten

Bei der Annotation der Dokumente wurden die Entitäten wie in Tabelle 3.4 spezifiziert händisch erfasst. Um die Aufgabe für den Experten zu erleichtern, wurde das Korpus durch die SemantLink Applikation vorab annotiert. Die Aufgabe des Experten war somit hauptsächlich die manuelle Verbesserung der Ergebnisse.

5.1.2 Performanzmaße

Als Performanzmaß dient das sogenannte *F-Maß*, welches aus einem gewichteten harmonischen Mittel[4] aus Genauigkeit (precision) und Trefferquote (recall) berechnet wird.

In Bereich der Informationsextraktion (IE) wird im Gegensatz zum Information Retrieval (IR) eine eigene Terminologie verwendet. Siehe Tabelle 5.1. Diese Unterschiede beruhen darauf, dass sich die Aufgaben und Ziele beider Disziplinen unterscheiden. Während die IE Community von „correct“ (korrekte Annotationen), „missing“ (fehlende Annotationen), „spurious“ (falsche Annotationen) und „partially correct„ (korrek aber falsche Spannweite der Annotation) spricht, werden in der IR Community folgende Begriffe verwendet: „true positives“ (korrekt Annotationen), „false negatives“ (fehlende

[2] http://en.wikipedia.org/wiki/Message_Understanding_Conference
[3] http://de.wikipedia.org/wiki/Kleine_und_mittlere_Unternehmen
[4] http://de.wikipedia.org/wiki/Harmonisches_Mittel

Tabelle 5.1: Vergleich der unterschiedlichen Terminologie in IE, IR und Inter Annotator Agreement[20]

Gold Standard (IR)	**Gold Standard (IE)**	**Inter Annotator Agreement**
Correct	True Positive	Match
Missing	False Negative	Only A (or B)
Spurious	False Positive	Only B (or A)
Partially Correct	-	Overlap

Annotationen) und „false positives“ (falsche Annotationen). Während die Gold Standard Vergleiche in IE und IR immer von einem korrekten Korpus ausgehen, werden beim Inter Annotator Agreement zwei Korpora miteinander verglichen ohne die Annahme, dass eines der beiden korrekter ist als das andere.

Genauigkeit (Precision)

Die Genauigkeit (Precision) beschreibt wie viele der Entitäten, welche durch die Applikation gefunden wurden, korrekt sind.

$$Precision = \frac{Correct}{Correct + Spurious} \tag{5.1}$$

Trefferquote (Recall)

Die Trefferquote (Recall) beschreibt wie viele der tatsächlichen Entitäten durch die Applikation gefunden wurden.

$$Recall = \frac{Correct}{Correct + Missing} \tag{5.2}$$

F-Maß

Precision und Recall tendieren dazu in einem sich gegenseitig beeinflussenden Verhältnis zu stehen [20, Vgl.]: Durch eine Verfeinerung der Regeln kann so zum Beispiel die Precision verbessert werden, gleichzeitig kann der Recall durch diese Maßnahme sinken. Von der anderen Seite betrachtet heißt dies, wenn man die Regeln sehr generell hält, dass man einen guten Recall Wert aber einen niedrigen Precision Wert erhält. Diese Tatsache macht den Vergleich der Performanz von Applikationen sehr schwierig. Aus diesem Grund wird das F-Maß als kombiniertes harmonisches Mittel der beiden Werte dafür eingesetzt.

Tabelle 5.2: Vergleich der unterschiedlichen Berechnung des F-Maß

Option	Beschreibung
Strict	Nur genau passende Annotationen werden als korrekt gewertet
Lenient	Auch teilweise korrekte (partially correct) Annotationen (falsche Spannweite) werden als korrekt gewertet
Average	Strict und Lenient Werte sind mit einem Mittel angenähert

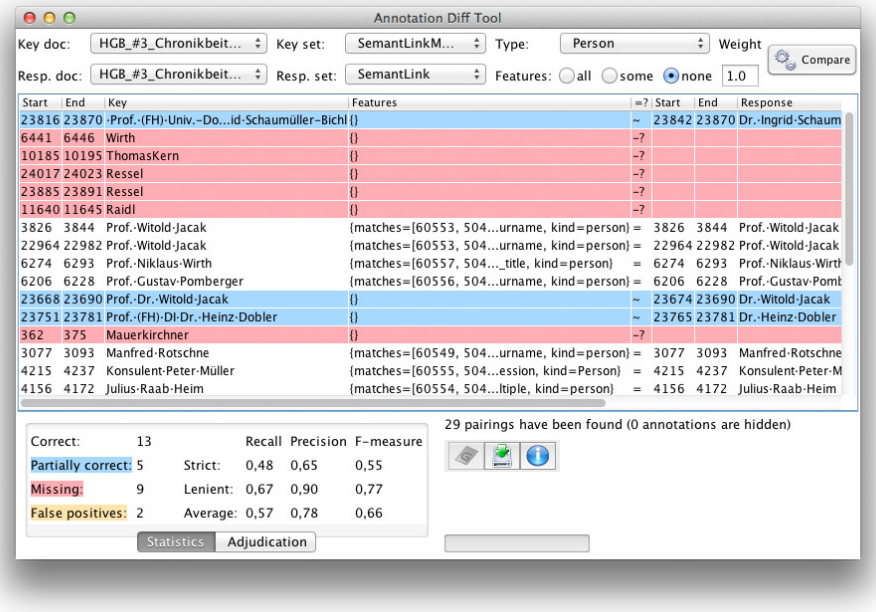

Abbildung 5.1: Screenshot des GATE Developer Annotation Diff Werkzeugs

$$F = 2 * \left(\frac{Precision * Recall}{Precision + Recall} \right) \tag{5.3}$$

5.1.3 Ablauf der Evaluierung

Die Evaluierung wurde auf Dokumentenebene mit dem in der Entwicklungsumgebung GATE Developer integrierten „Annotatioan Diff“ Werkzeug durchgeführt. Siehe Abbildung 5.1. Dieses Werkzeug ermöglicht das automatisierte Vergleichen von Annotationen in Dokumenten sowie das Berechnen des F-Maßes. Zur Berechnung werden drei verschiedene Optionen angeboten: Strict, Lenient und Average (Siehe Tabelle 5.2).

Das Corpus Quality Assurance Werkzeug weitet die Funktionsweise des

Tabelle 5.3: Liste an Dokumenten

Korpus Name	Inhalte	Dokumente
Korpus_Firmeninfo	Allgemeine Firmeninformationen	30
Korpus_Firmen_Presse	Firmen-Pressemitteilungen	25
Korpus_FE_Mail	E-Mails der F&E-GmbH und dazugehörige Anhänge	24
Korpus_FHOOE	Informationen über die FH-OOE	5
Korpus_All	**Gemischt**	**92**

Annotation Diff Werzeugs von einem Dokument auf einen ganzen Korpus aus. Zu beachten ist dabei, dass dieses Werkzeug dazu entwickelt wurde um das F-Maß basierend auf dem Inter Annotator Agreement (IAA) zu berechnen. Somit wird keine Aussage darüber getroffen, welches der zu vergleichenden Korpora der Gold Standard ist, sie werden nur als Korpus „A“ und „B“ gekennzeichnet.

5.1.4 Ergebnisse

Die Evaluierung basiert auf dem gesamten Korpus (5.3), welches 92 Dokumente umfasst. Im Nachfolgenden wird der manuell annotierte Gold Standard als Korpus „A“ bezeichnet. Das jeweils zu vergleichende Korpus wird als Korpus „B“ bezeichnet.

Die SemantLink Applikation wird nachfolgend als „SemantLink“ bezeichnet und mit der ANNIE Baseline Implementierung (nachfolgend „ANNIE“) sowie mit einer für die deutsche Sprache adaptierten Version von ANNIE verglichen (nachfolgend „ANNIE_GER“).

Es wurden drei Formen des F-Maß berechnet: Strict (F1-s), Lenient (F1-l) sowie Average (F1-a). Siehe Tabelle 5.2. Die Berechnung wurde sowohl auf Ebene der Entitätentypen als auch auf dem gesamten Korpus durchgeführt. Bei letzterer Berechnung wurde eine „Micro Summary“ berechnet, welche das gesamte Korpus als einziges großes Dokument betrachtet. Eine alternative Betrachtung wäre die „Macro Summary“, welche einen Durchschnitt aus den Ergebnissen der einzelnen Dokumente bildet. Kürzere Dokumente würden somit mehr Einfluss in die Berechnung haben. Da dies allerdings in diesem Fall nicht gewünscht ist, wird nur die Micro Summary berechnet.

Ergebnisse ANNIE

Die ANNIE Applikation performt mit einem F-Maß von 0,36 (strict) sehr mittelmäßig. Während der Precision Wert mit 0,50 im Mittelfeld liegt, ist

Tabelle 5.4: ANNIE: Ergebnisse der ANNIE Applikation (Baseline).

Annotation	Match	A	B	Overl.	Prec.	Rec.	F1-s	F1-l	F1-a
Date	280	166	138	180	0,47	0,45	0,46	0,75	0,60
Number	900	598	100	0	0,99	0,8	0,89	0,89	0,89
Money	10	91	23	1	0,29	0,10	0,15	0,16	0,15
Jobtitle	8	256	12	5	0,32	0,03	0,05	0,09	0,07
Person	284	139	435	93	0,35	0,55	0,43	0,57	0,50
Organization	309	575	927	168	0,22	0,29	0,25	0,39	0,32
Location	135	670	26	1	0,83	0,17	0,28	0,28	0,28
Micro Sum.	**1926**	**2495**	**1661**	**448**	**0,50**	**0,34**	**0,36**	**0,45**	**0,40**

der Recall (0,34) Wert sehr niedrig, was darauf schließen lässt, dass viele Annotationen gar nicht erkannt werden. Der Grund dafür liegt darin, dass ANNIE speziell für die englische Sprache entwickelt wurde und somit viele deutschsprachige Entitäten nicht erkannt werden. Einzig bei den sprachunabhängigen Nummern Annotationen wird ein F-Maß von 0,89 erreicht. Besonders niedrig sind die Werte bei den Entitäten vom Typ Money (Geldbeträge), Jobtitle (Beruf), Organizations (Firmen) und Locations (Orte). Viele kleine Firmen und Unternehmensbezeichnungen wie „GmbH“ werden nicht korrekt erkannt. Dasselbe trifft für Orte im deutschsprachigen Raum sowie für spezielle Berufsbezeichnungen und Geldbeträge zu. In Summe konnten 1926 Entitäten von insgesamt 6626 korrekt erkannt werden. Die Ergebnisse sind in Tabelle 5.4 zu sehen.

Ergebnisse ANNIE GER

Durch Adaptionen an der ANNIE Applikation für die deutsche Sprache konnten schon deutliche Verbesserungen festgestellt werden um die eben erwähnten Schwachstellen auszugleichen. Das F-Maß konnte von 0,36 auf 0,60 (strict) verbessert werden. Dies ist vor allem auf einen deutlich höheren Precision Wert von 0,80 zurückzuführen. Auch der Recall Wert konnte von 0,34 auf 0,48 erhöht werden. In Summe werden somit mit 2433 korrekt erkannten Annotationen deutlich mehr erfasst als bei der ANNIE Applikation ohne deutschsprachige Anpassungen. Die Ergebnisse sind in Tabelle 5.5 zu sehen.

Ergebnisse SemantLink

Die SemantLink Applikation, welche eine Kombination aus einem für die deutsche Sprache optimierten Gazetteer sowie einem dynamischen DBpedia Gazetteer besteht, konnte die Ergebnisse der ANNIE GER Applikation nochmals übertreffen. Durch die drastische Vergrößerung des Gazetteers konnten insgesamt 2870 Entitäten korrekt erkannt werden. Dies macht sich vor allem in einem höheren Recall Wert von 0,57 bemerkbar und in einem F-Maß von 0,65 (strict). Durch die enorme Vergrößerung des Gazetteers wurde zu Be-

Tabelle 5.5: ANNIE GER: Ergebnisse der ANNIE Applikation mit deutschsprachigem Gazetteer.

Annotation	Match	A	B	Overl.	Prec.	Rec.	F1-s	F1-l	F1-a
Date	291	259	138	76	0,58	0,46	0,51	0,65	0,58
Number	1353	328	13	0	0,99	0,80	0,89	0,89	0,89
Money	17	66	1	19	0,46	0,17	0,24	0,52	0,38
Jobtitle	49	213	43	7	0,49	0,18	0,27	0,30	0,29
Person	189	301	42	26	0,74	0,37	0,49	0,56	0,52
Organization	160	889	73	3	0,68	0,15	0,25	0,25	0,25
Location	364	430	157	12	0,68	0,45	0,54	0,56	0,55
Micro Sum.	**2433**	**2486**	**467**	**143**	**0,80**	**0,48**	**0,60**	**0,64**	**0,62**

Tabelle 5.6: SemantLink: Ergebnisse der SemantLink Applikation mit deutschsprachigem Gazetteer und externen Datenquellen.

Annotation	Match	A	B	Overl.	Prec.	Rec.	F1-s	F1-l	F1-a
Date	291	259	107	76	0,61	0,46	0,53	0,67	0,60
Number	1364	317	13	0	0,99	0,81	0,89	0,89	0,89
Money	18	65	1	19	0,47	0,18	0,26	0,53	0,39
Jobtitle	74	179	50	16	0,53	0,28	0,36	0,44	0,40
Person	331	137	58	48	0,76	0,64	0,69	0,80	0,75
Organization	191	834	210	27	0,45	0,18	0,26	0,29	0,28
Location	591	203	229	12	0,71	0,73	0,72	0,74	0,73
Micro Sum.	**2870**	**1994**	**668**	**198**	**0,77**	**0,57**	**0,65**	**0,70**	**0,67**

ginn der Precision Wert neben einem noch höheren Recall Wert stark nach unten gedrückt. Nach einer Verfeinerung der Entitätenregeln und Grammatiken konnte jedoch wieder ein beachtlicher Precision Wert von 0,77 erreicht werden, welcher nur knapp unter jenem der ANNIE GER Applikation (0,80) liegt. Die Ergebnisse der SemantLink Applikation sind in Tabelle 5.6 zu sehen.

Ergebnisse im Vergleich

Ein Vergleich der Ergebnisse der einzelnen Applikationen ist in Tabelle 5.7 zu finden. Hier wird die graduelle Verbesserung der einzelnen Werte klar ersichtlich. Während der Unterschied zwischen ANNIE und ANNIE GER klar ersichtlich ist, liegen die F-Score Werte von ANNIE GER und Semantlink mit 0,05 Prozentpunkten knapp beinander. Eine genauere Betrachtung der Recall und Precision Werte gibt aber Aufschluss darüber, dass eindeutig mehr Entitäten korrekt erkannt wurden, es aber auch einige Fehlannotationen (Missing und Spurious) gibt. Eine Visualisierung der Ergebnisse ist in Abbildung 5.2 zu sehen. Hier ist die graduelle Verbesserung in fast allen Bereichen klar ersichtlich. Einzig beim Precision Wert ist eine Verschlechterung von der ANNIE GER Applikation zur SemantLink Applikation erkennbar.

Tabelle 5.7: Ergebnisse der Applikationen (ANNIE, ANNIE GER und SemantLink) im Vergleich

Applikation	Match	A	B	Overl	Prec.	Rec.	**F1-s**	**F1-l**	**F1-a**
ANNIE	1926	2495	1661	448	0,50	0,34	**0,36**	**0,45**	**0,40**
ANNIE GER	2433	2486	467	143	0,80	0,48	**0,60**	**0,64**	**0,62**
SemantLink	2870	1994	668	198	0,77	0,57	**0,65**	**0,70**	**0,67**

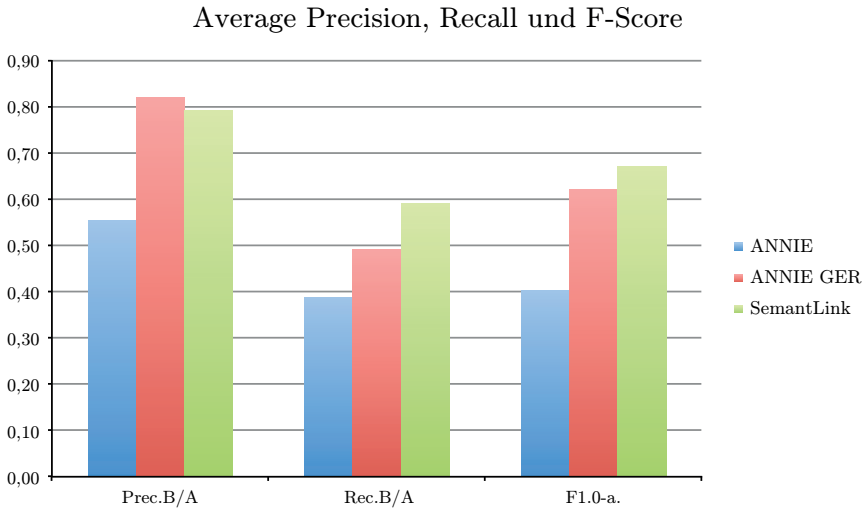

Abbildung 5.2: Durchschnittliche Precision, Recall und F-Score Werte

Bei einem Vergleich der F-Maße auf Entitätentyp Level sieht man, welche Maßnahmen bei welchen Entitätentypen besonders viel bringen: Aufgrund der Unterstützung von internationalen Datum und Nummern Formaten in allen Applikationen werden bei *Date* und *Number* die selben Ergebnisse erzielt. Bei *Money* machen sich speziell die deutsprachigen Entitätenregeln bemerkbar, während DBpedia hier keine weitere Verbesserung bringt. Bei den Berufsbezeichnungen und den Orten ist eine klare Verbesserung sowohl durch die deutschprachigen Regeln als auch durch die Verwendung des DBpedia Gazetteer zu sehen. Da die im Korpus verwendeten Personen und Organisationen jedoch kaum in DBpedia vertreten sind, liegen die Ergebnisse bei diesen Entitätentypen sehr knapp beisammen. Bei den Organisationen konnte sogar die ANNIE Baseline Applikation das beste Ergebnis erzielen. Dies liegt daran, dass es in diesem Bereich zu einer Vielzahl von Fehlannotationen kommt. Eine Visualisierung dieses Sachverhalts ist in Abbildung 5.3 zu sehen.

Für die Ergebnisse der SemantLink Applikation wurde zusätzlich noch der F-Score der einzelnen Dokumente betrachtet. Neben einigen Ausreißern, bei welchen es sich um extrem kurze Dokumente mit nur wenigen Entitäten

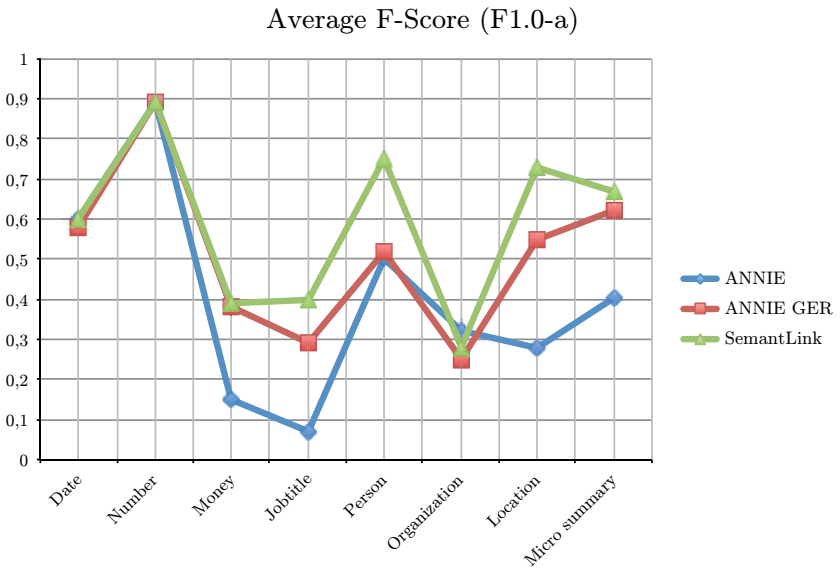

Abbildung 5.3: Average F-Score (F1.0-a)

aus dem E-Mail Korpus handelt, konnte ein konstanter F-Score zwischen 0,5 und 0,8 erreicht werden. Die Visualisierung 5.4 soll Aufschluss geben, bei welchen Dokumenten es noch Optimierungspotential gibt. Eine Optimierung auf dieses Korpus hin wäre somit denkbar, aber bei einem generischen Einsatz der Applikation nicht immer wünschenswert, da sich gewisse Verbesserungen in anderen Fällen oft wieder durch Fehlannotationen und somit einem geringeren Precision Wert bemerkbar machen.

Eine weitere wichtige Bewertungsgrundlage ist neben dem F-Maß auch die Ausführungszeit der Applikation. Gerade bei einer Anwendung auf ein größeres Korpus können sich kleine Unterschiede stark bemerkbar machen. Während bei den beiden Applikationen ANNIE und ANNIE GER, welche beide nur statische Listen und Regeln verwenden, die Ausführungszeit pro Dokument bei 0,25 - 0,26 Sekunden liegt wurde bei der SemantLink Applikation eine Ausführungszeit von 0,85 Sekunden pro Dokument gemessen. In Anwendung auf das gesamte Testkorpus von 92 Dokumenten ist dieser Unterschied bereits deutlich merkbar. Während ANNIE und ANNIE GER jeweils 23 bis 24 Sekunden für die Extraktion benötigen, dauert der Prozess bei der SemantLink Applikation 77,76 Sekunden. In dieser Hinsicht ist allerdings zu bemerken, dass es sich hier trotzdem um ein beachtliches Ergebnis bei der Verwendung von externen Datenquellen handelt und sämtliche in dieser Arbeit vorgestellten Optimierungsmöglichkeiten genutzt wurden um diesen Wert zu erreichen.

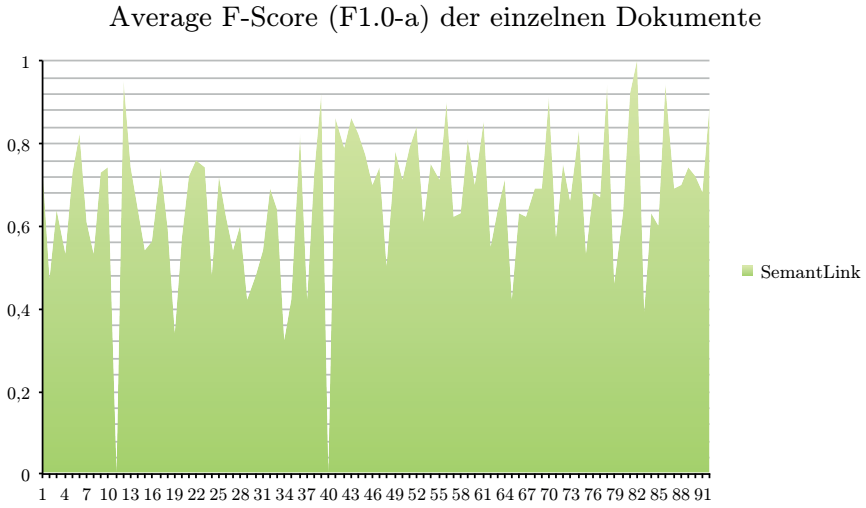

Abbildung 5.4: Average F-Score (F1.0-a) der einzelnen Dokumente

Tabelle 5.8: Ausführungszeit der Applikationen im Vergleich

Applikation	**Ausführungszeit Korpus (92)**	**Ausführungszeit/Dokument**
ANNIE	22.98	0,25
ANNIE GER	24.16	0,26
SemanLink	77,76	0,85

5.2 Relevanzbewertung

Da es sich bei der Bewertung der Relevanz um einen sehr spezifischen Anwendungsfall handelt, welcher in der Literatur in dieser Form noch nicht behandelt wurde, werden die Ergebnisse anhand eines Beispieldokuments erläutert. Auf einen Vergleich mit anderen Algorithmen wird aufgrund der fehlenden Gleichartigkeit verzichtet.

Als Beispieldokument wurde die deutschsprachige Wikipedia Seite über Barack Obama[5] verwendet. Die Global- und Kontextrelevanzen wurden in diesem Fall für die Entitätentypen Personen und Organisationen berechnet. Die Entscheidung, für welche Entitätentypen die Kontextrelevanz berechnet wird, ist dem Anwender überlassen. Zu beachten ist allerdings, dass aufgrund von technischen Einschränkungen ein Batch-Request für die Abfrage von verwandten Suchbegriffen mit dem TargetingIdeaSelector der Google AdWords API noch nicht möglich ist [6] und die Ausführungszeit somit stark von der Anzahl der Entitäten im Dokument abhängig ist.

Die Top 50 Ergebnisse pro Entitätentyp sind in Tabelle 5.9 nach abstei-

[5]http://de.wikipedia.org/wiki/Barack_Obama
[6]https://groups.google.com/forum/#!topic/adwords-api/r0GWYLrHnN8

gender Kontextrelevanz aufgelistet. Bei näherer Betrachtung lässt sich feststellen, dass die beiden mit Abstand relevantesten Personenentitäten Barack Obama und Obama sind. Beide weisen eine Kontextrelevanz von 1.0 auf. Barack Obama ist in diesem Fall zugleich auch die Personenentität mit der höchsten Globalrelevanz. Gereiht nach der Kontextrelevanz folgen weniger zu erwartende Personen, wie Abraham Lincoln oder Donald Trump. Der Grund für die hohe Bewertung ist, dass Barack Obama zum einen seit längerer Zeit mit Abraham Lincoln verglichen wird und dass dieses Thema gerade in Hinsicht auf die bevorstehende Wahl wieder aufgegriffen wird und somit einen hohen Aktualitätsbezug hat.[7] Donald Trump ist speziell seit Beginn des Wahlkampfes 2012 als einer der größten Unterstützer von Obamas Gegner Mitt Romney mit auffälligen Aussagen über Barack Obama aufgefallen.[8] Gefolgt werden diese Entitäten von Personen wie George W. Bush[9] und Martin Luther King[10]. Bei Betrachtung der Globalrelevanz sieht man, dass die Verteilung hier viel gleichmäßiger ist, da es sich in diesem Fall um global bekannte Personen handelt. Neben Abraham Lincoln, scheinen auch Obamas Frau Michelle Obama und der sudanesische Präsident Omar al-Bashir[11] auf.

Bei den Organisationsentitäten führt Ford die ebenfalls nach Kontextrelevanz gereihte Liste vor Spiegel, Die Zeit, YouTube, American, BP und Reuters an. In diesem Fall sind die Suchgewohnheiten der User besonders deutlich zu beobachten: Während YouTube, Die Zeit oder Reuters als Organisationen keinen allzugroßen direkten Bezug zu Barack Obama haben, suchen viele User nach z.B. dem YouTube Channel von Barack Obama[12]. Neben der Vielzahl an Medienunternehmen ist BP mit 0,60% an 6. Stelle im Ranking. Der Grund dafür ist die durch BP ausgelöste Ölpest im Golf von Mexiko 2010 und im Rahmen der Wahl wieder aufgetauchte Gerüchte über Verbindungen zwischen BP und Obama[13]. Das Ranking nach Globalrelevanz wird von Ford (0,89) vor YouTube (0,83%), MTV (0,78%), American (0,77%), Al Jazeera (0,70%), Spiegel (0,70%) und FAZ (0,68%) angeführt.

Abbildung 5.5 zeigt die Verteilung der Entitäten durch eine zwei-dimensionale Positionierungen anhand der Globalrelevanz (X-Werte) und der Kontextrelevanz (Y-Werte). Eine Besonderheit ist, dass sich trotz Normalisierung der Ergebnisse immer noch viele Entitäten im unteren Drittel befinden. Der Grund dafür ist, dass im evaluierten Dokument extrem starke Entitäten wie Barack Obama enthalten sind (28.952.913 Gefällt mir[14], 1.830.000

[7] http://www.huffingtonpost.com/2012/07/11/obama-lincoln_n_1663895.html

[8] http://www.csmonitor.com/USA/DC-Decoder/Decoder-Wire/2012/0530/Is-Donald-Trump-secretly-supporting-President-Obama

[9] http://www.thedailybeast.com/articles/2012/04/03/george-w-bush-barack-obama-s-best-friend-in-the-2012-election.html

[10] http://english.alarabiya.net/articles/2012/09/26/240283.html

[11] http://en.wikipedia.org/wiki/Omar_al-Bashir

[12] http://www.youtube.com/user/BarackObamadotcom

[13] http://www.politico.com/news/stories/0510/36783.html

[14] https://www.facebook.com/barackobama

Tabelle 5.9: Top 50 Personen und Organisationen Entitäten sortiert nach absteigender Kontextrelevanz.

Person	**Global**	**Kontext**	**Organisation**	**Global**	**Kontext**
Barack Obama	0,80	1.00	Ford	0,89	0,93
Obama	0,61	1.00	Spiegel	0,70	0,72
Abraham Lincoln	0,63	0,56	Die Zeit	0,59	0,69
Donald Trump	0,63	0,54	YouTube	0,83	0,66
George W. Bush	0,57	0,34	American	0,77	0,62
Than A	0,48	0,27	BP	0,65	0,60
Omar	0,58	0,26	Reuters	0,64	0,59
Stanley A	0,33	0,26	Al Jazeera	0,70	0,59
Joshua	0,48	0,25	Spiegel Online	0,64	0,59
John F	0,48	0,24	MTV	0,78	0,59
Luther King	0,51	0,24	FAZ	0,68	0,59
Martin Luther King	0,51	0,24	Die Welt	0,58	0,58
Barack	0,59	0,24	Washington Post	0,64	0,56
Osama	0,48	0,23	Wall Street	0,65	0,56
Jeremiah	0,35	0,22	The Economist	0,66	0,56
Van Buren	0,37	0,22	Declare	0,60	0,55
George W	0,51	0,22	Boston University	0,58	0,55
Sarah Palin	0,53	0,22	dpa	0,60	0,54
Tom Hanks	0,49	0,22	Foreign Affairs	0,58	0,54
Michelle Obama	0,55	0,22	Handelsblatt	0,55	0,54
Osama bin Laden	0,41	0,22	USA Today	0,61	0,54
Bill Clinton	0,48	0,21	Newsweek	0,61	0,54
Hillary Clinton	0,42	0,21	Accept	0,60	0,54
Angela Merkel	0,44	0,21	Reaktor	0,47	0,54
Glenn Beck	0,51	0,21	RAWA	0,58	0,53
Theodore Roosevelt	0,38	0,21	Human Rights Watch	0,61	0,53
Al Qaida	0,34	0,21	Tapper	0,51	0,52
Ivy League	0,35	0,21	Int. Herald Tribune	0,54	0,52
Manya A	0,32	0,21	Atomkraft	0,26	0,52
Hosni Mubarak	0,34	0,20	Regierung	0,39	0,51
Woodrow Wilson	0,36	0,20	Innenministerium	0,33	0,49
Jimmy Carter	0,36	0,20	Weltmacht	0,48	0,47
Francis of Assisi	0,33	0,20	Klinik	0,51	0,41
Xavier University	0,39	0,20	Facebook	0,67	0,35
Blagojevich	0,33	0,19	UN	0,60	0,30
Joe Biden	0,45	0,19	Wikipedia	0,54	0,30
John Roberts	0,35	0,19	A Story	0,62	0,29
Paul M	0,50	0,19	United	0,62	0,29
George H	0,48	0,19	BBC	0,55	0,29
Grant Park	0,33	0,19	Don	0,59	0,29
Jesse Jackson	0,32	0,19	EU	0,54	0,28
John McCain	0,47	0,19	The Week	0,50	0,27
Charles W	0,45	0,19	BBC News	0,49	0,27
Rick Warren	0,44	0,19	York City	0,57	0,26
John Boehner	0,44	0,18	The Guardian	0,13	0,26
John Kerry	0,36	0,18	The New York Times	0,53	0,26
Bill Text	0,34	0,18	CBS	0,48	0,26
Frank Act	0,27	0,18	Der Spiegel	0,42	0,26
Harold Washington	0,31	0,18	University of Michigan	0,48	0,25
Ted Kennedy	0,39	0,18	Wright	0,37	0,25

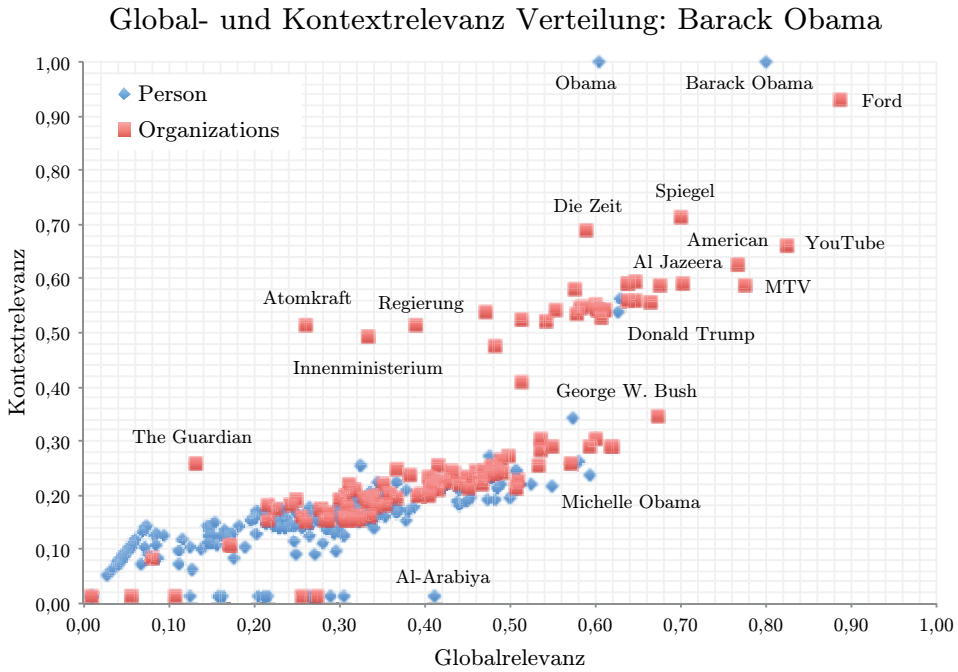

Abbildung 5.5: Global- und Kontextrelevanz Verteilung: Barack Obama

globale monatlich Suchanfragen).

Zusammenfassend lässt sich feststellen, dass sowohl die Bewertung der Globalrelevanz als auch der Kontextrelevanz sehr gute und aktuelle Ergebnisse liefert, welche durch statische Listen oder Kategorisierungen nie möglich wären. In diesen Fällen wären Personen wie Michelle Obama oder Organisationen wie das Weiße Haus an der Spitze der Rankings zu erwarten.

Kapitel 6

Schlussbemerkungen

6.1 Fazit

In der vorliegenden Arbeit wurde die Verwendung von Daten aus dem *Semantic-* und *Social Web* in Applikationen der Computerlinguistik evaluiert und zwei konkrete Anwendungsfälle vorgestellt. Zum einen handelt es sich dabei um die Performanzoptimierung einer bestehenden Applikation (der Entitätenextraktion) und zum anderen um die Vorstellung einer bisher noch neuen Anwendung zur Relevanzbewertung von Entitäten.

Die Implementierung dieser Ansätze im Rahmen der SemantLink Applikation hat gezeigt, dass sowohl eine Performanzsteigerung der Entitätenextraktion als auch eine sinnvolle Relevanzbewertung möglich sind. Das World Wide Web kann somit als computerlinguistische Ressource nicht nur als Korpus zur Analyse, sondern auch durch die daraus resultierende kollektive Intelligenz zur Verbesserung der Methoden der Computerlinguistik selbst herangezogen werden. Stets zu beachten sind allerdings die im Bereich der externen Datenquellen auftretenden Limitationen wie eingeschränkte Zugriffe oder Ausführungszeiten.

Diese Limitationen stellten für mich während des Evaluierens von konkreten Anwendungsbeispielen und deren Implementierung eine besondere Herausforderung dar. Oft konnten Probleme erst dann gelöst werden, wenn Änderungen an der verwendeten API durchgeführt wurden. So hat zum Beispiel die Einführung von Batch-Requests für die Facebook OpenGraph API zu einer enormen Minimierung der Ausführungszeit bei der Berechnung der Globalrelevanz geführt.

Zusammenfassend lässt sich sagen, dass gerade diese Herausforderung das Schreiben dieser Arbeit und das Entwickeln der SemantLink Applikation spannend und inspirierend gemacht haben, auch wenn teilweise in langen Nächten nach den Lösungen gesucht wurde.

6.2 Ausblick

Die in dieser Arbeit vorgestellen Implementierungsansätze müssen mit der Weiterentwicklung der verwendeten APIs und Datenquellen adaptiert werden. Während sich durch neue Funktionen neue Möglichkeiten ergeben können, kann es auch sein, dass einige Daten in Zukunft nicht mehr in der vorliegenden Form verfügbar sind. In diesem Fall muss nach alternativen Datenquellen gesucht werden.

Ein Optimierungspotential liegt in der Minimierung der Ausführungszeit der Relevanzbewertung im Speziellen der Kontextrelevanz, bei welcher aufgrund von Beschränkungen der Google AdWords API eine Bündelung der Abfragen noch nicht möglich ist. Weiters können die deklarierten Regeln und Grammatiken zur Entitätenextraktion stets weiter optimiert und entwickelt werden. Zu beachten ist dabei immer die Auswirkung auf das F-Maß eines passenden Korpus.

Neben den in dieser Arbeit vorgestellen Anwendungsfällen gibt es darüberhinaus eine Vielzahl weiterer potentieller Verwendungsmöglichkeiten von externen Datenquellen in der Computerlinguistik. Mit der steigenden Anzahl an immer neuen Daten und verbesserten Zugriffsmöglichkeiten werden sich in Zukunft noch viele weitere Möglichkeiten ergeben, welche zum Zeitpunkt des Verfassens dieser Arbeit noch nicht absehbar sind. Es lässt sich allerdings abschätzen, dass bei vielen heute noch aus statischen Listen, Datenquellen und Grammatiken bestehenden Komponenten der Computerlinguistik in Zukunft zunehmend externe Datenquellen aus dem World Wide Web eingesetzt werden.

Anhang A

Inhalt der CD-ROM

A.1 Masterarbeit (PDF)

Pfad: /

Masterarbeit.pdf	Entitätenextraktion und Relevanzbewertung mit Hilfe von semantischen Datenquellen und APIs.

A.2 SemantLink

Pfad: /SemantLink

SemantLink.zip	Generische Ressourcen des SemantLink GATE Plugins
SemantLinkAPI.zip . . .	JAVA Eclipse Projekt der SemantLink API
SemantLinkDeveloper.zip	JAVA Eclipse Projekt der SemantLink Applikation
SemantLinkLKB.zip . .	JAVA Eclipse Projekt des SemantLink LKB Gazetteer Plugins
SemantLinkRelevance.zip	JAVA Eclipse Projekt des SemantLink Relevance Plugins
SemantLinkAPI_Linux.war	SemantLink Webapplikation

A.3 Online-Quellen (PDF)

Pfad: /Onlinequellen

GATE_Module_2_IE.pdf	The University of Sheffield. Module 2: Introduction to IE and ANNIE
GNU_Lesser_General_Public_License (LGPL).pdf	GNU LESSER GENERAL PUBLIC LICENSE. 2012

Semantic_Web_W3C.pdf.pdf W3C - Semantic Web

The_Linking_Open_Data_cloud_diagram.pdf Linked Open Data Cloud Diagram

A.4 Evaluierung

Pfad: /Evaluierung

Korpus_All_SemantLinkManual-ANNIE.html Ergebnisse der ANNIE Entitätenextraktion

Korpus_All_SemantLinkManual-ANNIE_GER.html Ergebnisse der ANNIE_GER Entitätenextraktion

Korpus_All_SemantLinkManual-SemantLink.html Ergebnisse der SemantLink Entitätenextraktion

Quellenverzeichnis

Literatur

[1] Tim Berners-Lee. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*. HarperOne, 1999.

[2] Alessio Bosca und Luca Dini. „Automatic gazetteer generation from wikipedia“. In: *Proceedings of the 2009 international conference on Advanced language technologies for digital libraries*. NLP4DL'09/AT4DL'09. Springer-Verlag, 2011, S. 61–71.

[3] John G. Breslin, Alexandre Passant und Stefan Decker. *The Social Semantic Web*. Springer, 2009.

[4] Kai-Uwe Carstensen u. a. *Computerlinguistik und Sprachtechnologie: Eine Einführung*. 3. Aufl. Heidelberg: Spektrum Akademischer Verlag, 2009.

[5] Hamish Cunningham u. a. *Text Processing with GATE (Version 6)*. University of Sheffield, Departement of Computer Science, 2011.

[6] Gianluca Demartini u. a. „Semantically Enhanced Entity Ranking“. In: *Proceedings of the 9th international conference on Web Information Systems Engineering*. WISE '08. Springer-Verlag, 2008, S. 176–188.

[7] *Facebook Current Report, Form 8-K,* Techn. Ber. Menlo Park, California, 94025: Facebook Inc., 2012.

[8] Neumann Günther. *Informationsextraktion*. Techn. Ber. Kaiserslautern, Germany: DFKI GmbH, 2000.

[9] Mirella Lapata und Frank Keller. „Web-based models for natural language processing“. In: *ACM Trans. Speech Lang. Process.* 2.1 (Feb. 2005).

[10] Henning Lobin. *Computerlinguistik und Texttechnologie*. Stuttgart: UTB, 2009.

[11] John Makhoul u. a. „Performance Measures For Information Extraction“. In: *In Proceedings of DARPA Broadcast News Workshop*. 1999, S. 249–252.

[12] Diana Maynard, Kalina Bontcheva und Hamish Cunningham. „Automatic Language-Independent Induction of Gazetteer Lists“. In: *Proceedings of 4th Language Resources and Evaluation Conference (LREC 2004)*. 2004.

[13] Olena Medelyan u. a. „Mining meaning from Wikipedia“. In: *Int. J. Hum.-Comput. Stud.* 67.9 (Sep. 2009), S. 716–754.

[14] Pablo N Mendes u. a. „DBpedia Spotlight : Shedding Light on the Web of Documents“. In: *Proceedings of the 7th International Conference on Semantic Systems. I-Semantics 2011. Graz, Austria*. 2011.

[15] Marie-Francine Moens. *Information Extraction: Algorithms and Prospects in a Retrieval Context*. Dordrecht: Springer, 2006.

[16] Christof Müller und Iryna Gurevych. „Using Wikipedia and Wiktionary in domain-specific information retrieval“. In: *Proceedings of the 9th Cross-language evaluation forum conference on Evaluating systems for multilingual and multimodal information access*. CLEF'08. Springer-Verlag, 2009, S. 219–226.

[17] Simone Paolo Ponzetto und Roberto Navigli. „Knowledge-rich Word Sense Disambiguation rivaling supervised systems“. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. ACL '10. Association for Computational Linguistics, 2010, S. 1522–1531.

[18] Samuel Reese, Gemma Boleda und Montse Cuadros. „Wikicorpus : A Word-Sense Disambiguated Multilingual Wikipedia Corpus“. In: *Proceedings of 7th Language Resources and Evaluation Conference (LREC 2010)*. 2010, S. 1418–1421.

[19] Tomek Strzalkowski. *Natural Language Information Retrieval*. Springer, 1999.

[20] The University of Sheffield. *Module 2: Introduction to IE and ANNIE*. 2010. URL: http://gate.ac.uk/sale/talks/gate-course-may10/track-1/module-2-ie/module-2-ie.pdf.

[21] Katrin Tomanek und Udo Hahn. *The Muc7 Corpus*. Techn. Ber. Germany: Friedrich-Schiller-Universitat Jena, 2001.

[22] Anne-Marie Vercoustre, James A. Thom und Jovan Pehcevski. „Entity ranking in Wikipedia“. In: *Proceedings of the 2008 ACM symposium on Applied computing*. SAC '08. ACM, 2008, S. 1101–1106.

[23] Pu Wang u. a. „Using Wikipedia knowledge to improve text classification“. In: *Knowl. Inf. Syst.* 19.3 (Mai 2009), S. 265–281.

[24] Hugo Zaragoza u. a. „Ranking very many typed entities on wikipedia“. In: *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. CIKM '07. ACM, 2007, S. 1015–1018.

Online-Quellen

[25] *GNU LESSER GENERAL PUBLIC LICENSE*. 2012. URL: http://www.gnu.org/licenses/lgpl-3.0.de.html.

[26] *Linked Open Data Cloud Diagram*. 2011. URL: http://richard.cyganiak.de/2007/10/lod/.

[27] *W3C - Semantic Web*. 2012. URL: http://www.w3.org/standards/semanticweb/.