Extraktion semantisch relevanter Daten

aus natürlich sprachlichen Inhalten in Hinblick auf eine automatische Fragengenerierung

Masterarbeit

an der

Technischen Universität Graz

vorgelegt von

Joachim Weinhofer

April 2010

Betreuer: Univ.Doz. Dipl.-Ing. Dr. techn. Christian Gütl

Institut für Informationssysteme und Computer Medien (IICM),

Technische Universität Graz

A-8010 Graz

Extraction of relevant semantic data

from natural language texts
in the view of
automatic question generation

Master's Thesis

at the

Graz University of Technology

submitted by

Joachim Weinhofer

April 2010

Supervisor: Univ.Doz. Dipl.-Ing. Dr. techn. Christian Gütl

Institute for Information Systems and Computer Media (IICM),
Graz University of Technology
A-8010 Graz, Austria

Kurzfassung

In der heutigen globalisierten Welt der Informationsgesellschaft ist die immerwährende Weiterbildung jedes Einzelnen für ein Bestehen der Anforderungen in der Berufswelt nahezu unabdinglich. Für die Menschen ist es wichtig, gezielt an Information zu gelangen, welche diese Anforderungen erfüllt. Um nun genau jene Inhalte aus der unglaublichen Vielfalt an Information herausfiltern zu können, bedarf es Konzepten, welche die wesentlichen Inhalte aus diesen extrahieren und zugänglich machen. Natürlich wäre es auch wünschenswert, diesen Wissenszuwachs automatisch verifizieren zu können. An diesen beiden Punkten setzt das zentrale Thema dieser Arbeit an.

Im Zuge dieser Arbeit wurde ein System entwickelt, das aus natürlich sprachlichen Texten und Dokumenten die wesentlichen Inhalte extrahiert, diese dem Nutzer zugänglich macht und anschließend mit diesen extrahierten Daten Fragen generiert, welche eine Überprüfung des Verständnisses der essentiellen Inhalte des Ausgangsdokuments ermöglichen.

Dabei wurden intensive Recherchen in dem Gebiet der Linguistik und der Verarbeitung natürlicher Sprache durchgeführt, um grundlegende Kenntnisse in dieser Thematik zu erlangen. Anschließend wurden die Anwendungsgebiete in diesen wissenschaftlichen Bereichen, welche mit dem zentralen Thema dieser Arbeit in Verbindung stehen, analysiert und wichtige Erkenntnisse für die konkrete Umsetzung eines solchen Systems extrahiert. Zusätzlich wurde ein Überblick über den aktuellen Forschungstand in diesem Gebiet geboten.

Anhand dieser Untersuchungen wurde erkannt, dass sowohl statistische als auch semantische Methoden der Textanalyse zielführend sein können. Darüber hinaus zeigten viele Forschungsansätze, dass sich auch aus der Struktur der Dokumente wichtige Informationen bezüglich des Inhalts ableiten lassen. Daher wurden verschiedenste Arten dieser Analysemethoden in das entwickelte System zur Extraktion von relevanten Daten integriert.

Die anschließende Evaluierung des Concept Extractors verdeutlichte, dass das System Konzepte extrahieren kann, welche qualitätsmäßig durchaus mit manuell extrahierten Konzepten vergleichbar sind.

Abstract

In today's globalized world of the information society the continuing process of learning is almost indispensable for everybody to meet the challenges of the working world. For the people it is important to specifically get information which fulfills these requirements. To exactly get these contents from the great variety of information concepts are needed which extract these contents and make it available. Certainly it would also be desirable to be able to verify the amount of knowledge acquired automatically. These two issues are the central focus of this work.

In the course of this work a system was developed, which extract from natural language texts and documents the essential contents and makes these contents accessible for the user. Afterwards questions are generated based on these contents. These questions should make it possible to examine the understanding of the essential contents of the original text.

Thereby intensive inquiries were made about the fields of linguistics and natural language processing in order to gain basic knowledge in this topic. Subsequently, the areas of application within the scientific ranges which are closely related with the central theme of this work were analyzed in order to get important knowledge for the concrete implementation of such a system. In addition, an overview of the current state of research in this area was given.

Based on these studies it was recognized that both statistic and semantic methods of the text analysis may be useful. Furthermore many research approaches showed that important information concerning the content could also be derived from the structure of a document. Therefore, various types of analytical methods have been integrated in the developed system for extracting relevant data.

The following evaluation of the concept extractor clarified that the system can extract concepts, which are qualitatively quite comparable to manually extracted concepts.

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am …… ……..
Unterschrift

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…… ……..
date signature

Danksagung

Mein herzlicher Dank gilt meinen Eltern, Anita und Rudolf, die es mir ermöglicht haben, ein Studium zu verfolgen und mich immer in jeglicher Art und Weise unterstützt haben.

Ein besonderes Dankeschön gilt auch meinem Betreuer, Univ. Doz. Dipl.-Ing. Dr. techn. Christian Gütl, der mir die Möglichkeit bot, diese Arbeit zu verfassen und der mich während der gesamten Dauer immer umfassend betreute.

Darüber hinaus möchte ich noch meinem Studienkollegen Klaus Lankmayr recht herzlich danken, der seine Diplomarbeit der im gleichen Zeitraum wie ich selbst verfasst hat, und dessen Thema, die Automatische Fragengenerierung, sich ausgezeichnet mit meiner Arbeit ergänzt hat. Darüber hinaus danke ich Klaus für die gemeinsame Entwicklung von zentralen Elementen des Gesamtsystems, dem Automatic Question Creators.

Abschließend möchte ich mich bei Ass.-Prof. Mag. Dr.phil. Rudolf Muhr recht herzlich bedanken, der mir bei Unklarheiten bezüglich sprachrelevanter Themen immer geholfen hat und mir dabei viele nützliche Tipps gegeben hat.

Weinhofer Joachim
Graz, Österreich, April 2010

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Das Ziel dieser Arbeit ist es Methoden und Konzepte zu entwickeln, die eine maschinelle Verarbeitung natürlich sprachlicher Texte dahingehend ermöglichen, semantisch relevante Informationen aus diesen Texten im Hinblick auf eine automatische Fragengenerierung zu extrahieren. In diesem Kapitel wird einleitend kurz auf die Notwendigkeit eines derartigen Systems eingegangen und in weiterer Folge die Struktur dieser Arbeit näher erläutert.

1.1 Motivation und Überblick

In den letzten Jahren haben sich das Angebot von und die Nachfrage nach Information weitestgehend verändert. So ist es heutzutage unbedingt erforderlich, sich immerwährend fortzubilden um den Anforderungen des Alltages gerecht werden zu können. Dabei ist es nicht mehr das Problem, an die gewünschten Informationen zu kommen, sondern, ganz im Gegenteil dazu, aus der unglaublichen Informationsmenge die wichtigsten Inhalte herauszufiltern. Nach Jackson (2002) ist das Bedürfnis nach Wissen und Information und die Befriedigung dieses Bedürfnisses der am stärksten wachsende Markt der Welt. Dieses Wissen wird in der heutigen, technisch weit fortgeschrittenen, Welt häufig aus dem Internet bezogen.

Laut einem Bericht des Guardian (2010) ist die Datenmenge, welche das Internet beherbergt, mittlerweile auf fast 500 Milliarden Gigabyte angewachsen. Diese nahezu unglaubliche Menge an Daten wird sich dennoch in den nächsten 18 Monaten aller Voraussicht nach verdoppeln. Zusätzlich wird geschätzt, dass mehr als 80 der vorhandenen Information textuell vorhanden ist (Wilks und Catizone, 1999).

Um bei dieser rasanten Entwicklung der Informationsvielfalt nicht den Überblick zu verlieren, ist es notwendig, Technologien bereit zu stellen, die es ermöglichen, aus dieser riesigen Datenmenge spezifische Information herauszufiltern und zu präsentieren. Solche Technologien, wie Suchmaschinen, Information Retrieval Systeme usw., sind bereits sehr zahlreich vorhanden und liefern auch zufrieden stellende Ergebnisse.

Im Zuge dieses Informationsbeschaffens und der Informationsaneignung wäre es natürlich von Vorteil, wenn dieses erlernte Wissen auch automatisch verifizierbar ist. Dies ist der Punkt, an dem das zentrale Thema dieser Arbeit einhakt.

Ziel dieser Arbeit ist es Konzepte und Methoden zu finden und damit ein System zu entwickeln, welches aus natürlich sprachlichen Texten semantisch

relevante Daten extrahiert. Diese Daten sollen dabei die wesentlichsten Inhalte und Konzepte der Texte repräsentieren. Zusätzlich sollen diese extrahierten Konzepte als Grundlage für das automatische Generieren von Fragen dienen. Die dabei erzeugten Fragen sollen den Lernenden dabei helfen, die Informationen aufzunehmen bzw. das Gelernte zu überprüfen.

1.2 Struktur der Arbeit

Diese Arbeit gliedert sich im Wesentlichen in zwei Teile. Im ersten (theoretischen) Teil (Kapitel 2 – Kapitel 4) wird näher auf das Thema dieser Arbeit, die Hintergründe und aktuelle Forschungsansätze eingegangen. Der zweite (praktische) Teil (Kapitel 5 – Kapitel 7) befasst sich mit der Umsetzung der aus dem ersten Teil gewonnen Erkenntnisse in ein System für die Extraktion von semantisch relevanten Daten in Hinblick auf eine automatische Fragengenerierung.

In Kapitel 2 wird näher auf die grundlegenden Eigenschaften von Sprache und Schrift eingegangen. Darüber hinaus wird ein Einblick in die maschinelle Verarbeitung natürlicher Sprache gegeben. Dabei werden vor allem Mechanismen zur automatischen Analyse von Texten, bezüglich der Identifizierung von bestimmten Strukturen in diesen bzw. der Extraktion bestimmter Merkmale aus diesen, vorgestellt.

In Kapitel 3 wird ein Überblick über einige Teilgebiete des Natural Language Processing gegeben. Dabei wird das Hauptaugenmerk auf jene Teilgebiete gelegt, die mit der Extraktion von semantisch relevanten Daten in Verbindung stehen und für diese Aufgabe wichtige Erkenntnisse liefern.

Der aktuelle Forschungstand einiger der im Kapitel 4 vorgestellten Teilgebiete der maschinellen Verarbeitung natürlicher Sprache wird in diesem Kapitel aufgezeigt. Dabei richtet sich der Fokus auf jene Gebiete des NLP, die für die Extraktion semantisch relevanter Inhalte aus natürlich sprachlichen Daten von essentieller Bedeutung sind.

Die in den vorangegangen Kapiteln gewonnenen Erkenntnisse werden im Kapitel 5 dazu genutzt, um Konzepte zu generieren, die für ein System für die automatische Extraktion von semantisch relevanten Daten benötigt werden. Dieses Kapitel beinhaltet die Festlegung der an dieses System gestellten Anforderungen sowie das Aufzeigen jener Konzepte, welche die Umsetzung eines derartigen Systems ermöglichen.

In Kapitel 6 werden jene Tools und Frameworks vorgestellt, die bei der Realisierung der in vorangegangenen Abschnitt aufgestellten Konzepte erforderlich

sind. Dabei wird der Fokus hauptsächlich auf die NLP – Tools gelegt, die in weiterer Folge bei der Umsetzung des Systems verwendet werden.

In Kapitel 7 wird dann die konkrete Umsetzung jenes System vorgestellt, das den Anforderungen der Extraktion von semantisch relevanten Daten genügen soll. Dabei wird kurz auf die grundlegende Architektur eingegangen, sowie die konkrete Implementierung näher erläutert. Zusätzlich werden Probleme aufgezeigt, die während der Entwicklungsphase aufgetreten sind. Danach wird die Applikation und die Funktionalität, die das System bietet, näher erläutert, sowie Verbesserungs – und Erweiterungsmöglichkeiten aufgezeigt. Abschließend wird noch eine Evaluierung des Systems durchgeführt.

In Kapitel 8 werden dann die Erkenntnisse präsentiert, die der Autor während des Verfassens dieser Arbeit und der Entwicklung des Systems gewinnen konnte. Dabei wird sowohl auf die Erkenntnisse im Zuge des Implementierungsprozesses näher eingegangen, als auch andere Erfahrungen aufgezeigt, welche beim Verfassen dieser Arbeit gewonnen wurden.

Abschließend werden in Kapitel 9 die wesentlichsten Inhalte dieser Arbeit kurz zusammengefasst und Vorschläge aufgezeigt, wie der Concept Extractor weiter verbessert werden kann.

2 Einführung in die Verarbeitung natürlicher Sprache

Die Voraussetzung für die Extraktion semantisch relevanter Inhalte aus natürlich sprachlichen Dokumenten ist das Verständnis der Sprache und des Schriftbildes der Sprache dahingehend, Strukturen und Eigenschaften der Sprache bzw. des Schriftbildes zu erkennen, um aus diesen Erkenntnissen die richtigen Schlüsse zu ziehen und dementsprechend die essentiellen Inhalte aus diesen Dokumenten extrahieren zu können.

Um nun diese Strukturen und Eigenschaften erkennen zu können, ist es unabdinglich, die sprachlichen Grundlagen und Hintergründe zu kennen. Deshalb werden in diesem Abschnitt die Grundlagen näher erläutert, die für eine derartige Extraktion erforderlich sind. Dabei werden wichtige Begriffe der Linguistik und der Computerlinguistik vorgestellt, um die Bedeutung dieser wissenschaftlichen Teildisziplinen in Bezug auf das Thema dieser Arbeit hervorzuheben. Zusätzlich wird auf das Gebiet der Verarbeitung natürlicher Sprache näher eingegangen.

2.1 Grundlagen

In diesem Abschnitt werden wichtige Begriffe und Grundlagen der Linguistik erläutert, die mit dem Thema dieser Arbeit in Zusammenhang stehen und für das weitere Verständnis erforderlich sind. So wird in weiterer Folge näher auf die Unterschiede und Gemeinsamkeiten von Sprache und Schrift eingegangen und versucht daraus Eigenschaften und Strukturen des Schriftbildes abzuleiten, um ein automatisiertes semantisches Verstehen von Texten und Dokumenten zu ermöglichen.

2.1.1 Sprache und Schrift

Laut dem Bertelsmann Lexikon (1996a) ist Sprache ein Sammelbegriff für unterschiedliche Fähigkeiten und Sozialgebilde. Sprache definiert sich dabei als die allgemeine (menschliche) Fähigkeit des Zeichengebrauchs, um dabei Wahrnehmungen, Eindrücke, Gefühle, Phantasien, Erinnerungen und Ähnliches ausdrücken zu können. Sprache ist also ein System von Zeichen für Begriffe und Gegenstände in Verbindung mit strukturellen Regeln, die es möglich machen, die einzelnen Zeichen miteinander zu kombinieren (vgl. Bertelsmann, 1996a)

Die Schrift ist entsprechend Bertelsmann (1996b) ein System von Zeichen, welches menschliche Äußerungen und Begriffe zum Zwecke der Informationsvermittlung und Informationssicherung sichtbar macht. Schrift ist also die geschriebene Form der Sprache, obwohl zwischen Sprache und Schrift keine

notwendige natürliche Beziehung bestehen muss. So kann prinzipiell jede natürliche Sprache mit jeder Schrift niedergeschrieben werden (vgl. Bertelsmann, 1996b).

Schrift und Sprache haben allerdings eine Gemeinsamkeit, nämlich die Verwendung von Zeichen in Abhängigkeit von bestimmten Regeln, der sogenannten Grammatik (siehe 2.1.2). Diese regelt die Verwendung der Zeichen und deren Kombinationen und machen damit eine Kommunikation erst sinnvoll und möglich (vgl. Bertelsmann, 1996b).

In den weiteren Abschnitten dieser Arbeit wird Schrift, also die geschriebene Sprache, Text genannt. Text besteht aus einer Folge von Zeichen, welche aus einem, im Normalfall, begrenzten Zeichenvorrat entnommen werden. Dabei dürfen sich die Zeichen nicht gegensätzlich beeinflussen und zusätzlich müssen die einzelnen Wörter eindeutig voneinander abgegrenzt sein (vgl. Pfister, 2008).

2.1.2 Grammatik

Laut Volmert (2005) werden für den Begriff Grammatik drei Unterscheidungen getroffen. Einerseits ist Grammatik eine Sammlung von Regeln im Sinne von Vorschriften, die es ermöglichen, eine sogenannte Standardsprache fehlerfrei zu erlernen. Dies wird als präskriptive bzw. normative Grammatik bezeichnet und dient dazu, den Soll – Zustand einer Sprache zu beschreiben. Im Gegensatz dazu wird die Beschreibung des Ist-Zustandes einer Sprache, also alle feststellbaren und erfassbaren Regeln einer Sprache, als deskriptive Grammatik bezeichnet. Als dritte Unterscheidung nennt Volmert (2005) die innere Grammatik, welche das Regelsystem darstellt, das von einem Individuum in seiner Kindheit erworben und als Teil der Sprache von Generation zu Generation weitergegeben wird. Zusammenfassend kann man sagen, dass die Bedeutungen zwar wesentliche Unterschiede aufweisen, allerdings haben sie eines gemein: Sie sind prinzipiell Regelwerke. All diese Regelwerke und damit auch alle Teilgebiete der Grammatik befassen sich nach Wurzel (2000) mit der Struktur beziehungsweise mit der Form von sprachlichen Äußerungen. Der prinzipielle Unterschied zwischen diesen Regelwerken besteht nur in verschiedenen Sichtweisen, mit denen dieser Thematik begegnet wird (vgl. Volmert, 2005; Wurzel, 2000).

Für Kürschner (2007) hat Grammatik ebenfalls die Beschreibung der Sprachstruktur zur Aufgabe und er gliedert die Grammatik in die folgenden Teilgebiete:

2.1.2.1 Phonologie

Phonologie ist die Lehre von den Sprachlauten und beschäftigt sich mit der Aussprache und mit den Variationen von Lauten in Abhängigkeit verschiedenster Kontexte. Darüber hinaus befasst sich die Phonologie damit, wie sich die

verschiedensten Variationen der Aussprache der Laute auf andere Teilgebiete der Grammatik, wie zum Beispiel Morphologie und Syntax, auswirken (vgl. Walther, 2001).

Phonologie spielt in der Computerlinguistik, und damit in dem Bereich der Extraktion von semantisch relevanten Inhalten, nur eine untergeordnete Rolle, da dieses Teilgebiet der Linguistik hauptsächlich nur in den Bereichen der automatische Spracherkennung und der Sprachsynthese Anwendung findet (vgl. Hausser, 2000).

2.1.2.2 Graphematik

Die Graphematik versucht Verbindungen zwischen den phonologischen Einheiten und den Elementen der Schrift aufzustellen und untersucht die Kombinationsmöglichkeiten dieser. Ähnlich wie Phonologie spielt auch die Graphematik für die Computerlinguistik keine bedeutende Rolle (vgl. Kürschner, 2007).

2.1.2.3 Morphologie

Für Amtrup (2001) ist Morphologie die Beschreibung der Bildung und der Struktur von Wörtern. Hauptziel der Morphologie ist es, Grundformen von Wörtern unabhängig von ihren Flexionsformen zu ermitteln. Dabei wird versucht, Regeln und Bildungsgesetze zu finden, welche die Prozesse, die sich für die Bildung der verschiedensten Flexionsformen von Wörtern verantwortlich zeichnen, näher zu beschreiben. Darüber hinaus wird untersucht, wie sich die Verwendung und die Bedeutung der Wörter verändern, wenn diese Regeln und Bildungsgesetze auf diese Wörter angewandt werden.

Diesem Teilgebiet der Grammatik und deren Anwendungen kommen in der Linguistik eine große Bedeutung zu und finden vor allem bei stark flektierenden Sprachen wie der deutschen Sprache Verwendung, werden aber auch bei weniger stark flektierenden Sprachen wie dem Englischen eingesetzt. Im Gegensatz zu Graphematik und Phonologie spielt die Morphologie im Bereich der Computerlinguistik und auch im Bereich des Auffindens semantischer Informationen eine große Rolle. Daher wird im Zuge dieser Arbeit noch auf verschiedenste morphologischen Analyseverfahren näher eingegangen (vgl. Amtrup. 2001).

2.1.2.4 Wortartenlehre

In der Wortartenlehre wird das gesamte Reservoir von Wörtern einer Sprache in verschiedene Kategorien, in die sogenannten Wortarten, unterteilt. (vgl. Kürschner, 2007).

Hausser (2000) unterteilt die Wörter natürlicher Sprachen in folgende Wortarten:

- Substantive (Nomen), z.B.: Haus, Mensch, Tier etc.
- Verben, z.B.: gehen, sitzen, laufen etc.
- Adjektive, z.B.: groß, schön, dunkel etc.
- Adverbien, z.B.:, wo, danach, dort, hier etc.
- Artikel, z.B.: der, die, das, dem, den etc.
- Konjunktionen, z.B.: und, oder, aber, ob etc.
- Präpositionen, z.B.: an, nach, aus, bis etc.
- Partikel, z.B.: sehr, halt, ja, nein etc.

Die Klassen Substantive, Verben, Adjektive und Adverbien werden in der Linguistik auch Inhaltswörter genannt, die Wörter der anderen Klassen sind Funktionswörter. Dieser Umstand ist natürlich aus Sicht der Extraktion von semantisch relevanten Inhalten von großer Bedeutung, da sich somit die Bedeutung von komplexen Ausdrücken auf Bedeutung von einzelnen (Inhalts-) Wörtern zurückführen lässt. Allerdings ist der Begriff Inhaltswörter grundsätzlich nur auf Wörter anwendbar, die als Symbole bezeichnet werden können. Ein Wort ist dann ein Symbol, wenn sich das Wort ikonisch zu dem Bezeichneten verhält. Solche Wörter sind beispielsweise Wetter, Haus, Küche usw. Eine etwas detailliertere Beschreibung eines Symbols liefert der vierte Hauptsatz der Pragmatik (vgl. Hausser, 2000, Seite 115), worauf im Zuge dieser Arbeit allerdings nicht näher eingegangen wird (vgl. Hausser, 2000).

2.1.2.5 Syntax

Syntax ist die Lehre von der Komposition der Wortformen (vgl. Hausser, 2000). Laut Amtrup (2001) umfasst der Bereich der Syntax alle mit der Bildung und Struktur von Sätzen in Verbindung stehenden Komponenten. Für Hausser (2000) dient die Syntax als Grundlage für eine semantische Interpretation von Sprache und Schrift und ist somit Teil der Bedeutungslehre. Die grundlegende Aufgabe der Syntax ist infolgedessen, Wörter, zum Zwecke der Mitteilung von bestimmten Inhalten nach bestimmten Regeln, so anzuordnen, dass eine kontextspezifische Interpretation möglich ist. Erst über diese Interpretationsmöglichkeit definiert sich der eigentliche Sinn der Sprache, welcher darin besteht, die Kommunikation zwischen Individuen zu ermöglichen (vgl. Paul, 2000).

Grundvoraussetzung für eine erfolgreiche Kommunikation ist es, dass alle Kommunikationsteilnehmer die Aussagen des Sprechers auf dieselbe Art und Weise wie der Sprecher selbst interpretieren. Dieser Aspekt ist wichtig, da ein einzelnes Wort kontextabhängig sehr viele Bedeutungen haben kann. Erst durch eine

spezifische Wortumgebung und einen spezifischen Kontext bekommt auch das Ausgangswort einen speziellen Sinn, der im besten Falle auch eindeutig ist. Alle möglichen Umgebungen für ein Wort werden durch die Syntax spezifiziert, wobei es durchaus möglich sein kann, dass mehrere Wortumgebungen, sprich Sätze und Satzteile, im gleichen Kontext dieselbe Bedeutung haben. Es ist allerdings auch möglich, dass ein und dieselbe Wortumgebung in verschiedenen Kontexten verschiedene Bedeutungen haben (vgl. Paul, 2000).

Alleine dieser Umstand verdeutlicht bereits die hohe Komplexität des Themengebietes dieser Arbeit, da es sehr schwierig ist, solche kontextspezifischen Ambiguitäten richtig zu interpretieren und allgemein gültige Formalismen zu entwickeln, die diese geeignet beschreiben und damit auch der maschinellen Verarbeitung natürlicher Sprache zugänglich machen.

2.1.2.6 Textgrammatik

Die Textgrammatik untersucht im Gegensatz zu der Syntax keine Einzelsätze sondern ganze Texte, um unterschiedliche Textarten zu identifizieren und deren Merkmale zu finden. (vgl. Kürschner, 2007). Laut Gansel und Jürgens (2007) macht es die Textgrammatik, und damit eine Klassifizierung von Texten in Abhängigkeit bestimmter Merkmale nach spezifischen Textarten, erst möglich, diese Texte in richtiger Art und Weise zu interpretieren und zu verstehen.

2.1.2.7 Orthographie.

Orthographie wird auch Rechtschreibung genannt und ist laut Bertelsmann (1996c) definiert als die einheitliche, normierte Schreibung der Sprache. Die Rechtschreibung stellt eine Beziehung zwischen den Schriftzeichen und der Aussprache dieser Schriftzeichen bzw. der Aussprache der Kombinationen von Schriftzeichen her. Außerdem bestimmt die Rechtschreibung schreibspezifische Regeln wie zum Beispiel Groß - und Kleinschreibung, Zeichensetzung sowie Getrennt - und Zusammenschreibung (vgl. Kürschner, 2007). Dieses Teilgebiet der Grammatik ist für die Extraktion von semantisch relevanten Inhalten weniger von Bedeutung und daher wird auch nicht näher darauf eingegangen.

2.1.3 Semantik

Semantik ist definiert als die Lehre von der Bedeutung von Zeichen, hauptsächlich von Wörtern und Sätzen. In der Linguistik befasst sich die Semantik mit der Bedeutung und mit dem Inhalt von natürlich sprachlichen Ausdrücken und mit der Beziehung dieser Ausdrücke zu den Gegenständen und Begriffen, welche sie bezeichnen (vgl. Bertelsmann, 1996d).

Schiehlen (2001) unterteilt die Semantik in drei unterschiedliche Disziplinen und zwar in die lexikalische-, die Satz- und die Diskurssemantik. Die lexikalische Semantik befasst sich mit der Bedeutung von Wörtern währenddessen die Satzsemantik die Deutung von Sätzen zur Aufgabe hat. Die Diskurssemantik wiederum befasst sich mit der Interpretation und Bedeutung von ganzen natürlich sprachlichen Texten und Dokumenten (Schiehlen, 2001).

Die Entwicklung der modernen Semantik beruht auf zwei Grundprinzipien. Einerseits gibt es die formale Semantik, deren Konzepte darauf abzielen, die Beziehungen von Sätzen untereinander sowie deren Bedeutung mittels Logik und logischen Schlussfolgerungen zu erklären. Anderseits wird die moderne Semantik auch von Aspekten der Sprachphilosophie, wie Anaphorik, Präsuppositionen oder die Sprechtakttheorie, beeinflusst. Die diskursorientierte Semantik beruht hauptsächlich auf diesen Prinzipien (vgl. Schiehlen, 2001).

2.2 Natural Language Processing

Der Begriff Natural Language Processing (NLP, maschinelle Verarbeitung natürlicher Sprache) bezeichnet nach Jackson und Moulinier (2002) die Verarbeitung von natürlicher Sprache mithilfe von Computersystemen. Hierbei wird versucht gesprochene beziehungsweise geschriebene Sprache zu analysieren, um diese verstehen und auch reproduzieren zu können (Jackson & Moulinier, 2002)

Die maschinelle Verarbeitung natürlicher Sprache umfasst ein sehr umfangreiches Gebiet und findet in sehr vielen Bereichen des täglichen Lebens Anwendung. Sehr eng mit dem Gebiet der maschinellen Sprachverarbeitung verflochten ist die sogenannte Computerlinguistik. Diese beschäftigt sich ebenfalls mit der Verarbeitung natürlicher Sprache und ist zwischen den wissenschaftlichen Bereichen der Informatik und der Linguistik angesiedelt. Diese beiden Bereiche liefern grundsätzliche Erkenntnisse, welche benötigt werden, um Methoden zu entwickeln, die eine eigenständige, automatische bzw. semi-automatische maschinelle Verarbeitung von natürlicher Sprache ermöglichen (vgl. Carstensen et al., 2001).

Einige dieser Methoden und Verfahren der Textanalyse, die mit dem Thema dieser Arbeit in Verbindung stehen werden nachfolgend vorgestellt und näher erläutert. Dabei werden zuerst notwendige Vorverarbeitungsschritte aufgezeigt und danach einige Analyseverfahren präsentiert.

2.2.1 Text Preprocessing

Text Preprocessing ist ein sehr umfangreicher Vorverarbeitungsschritt und ist in nahezu jedem der Teilbereiche der maschinellen Verarbeitung natürlicher Sprache

unabdinglich. Ohne Vorverarbeitung der Daten, sprich des Textes beziehungsweise der Dokumente ist es beinah unmöglich, zufriedenstellend Ergebnisse zu erreichen. Die folgenden Methoden finden nahezu in allen Bereichen der Sprachverarbeitung Anwendung und werden daher nachfolgend genauer erläutert.

2.2.1.1 Tokenisierung

Unter Tokenisierung versteht man die Aufspaltung von Texten und Dokumenten in sogenannte Tokens, die eine sinnvolle Folge von Buchstaben und/oder Ziffern darstellen und eine gewisse semantische Information beinhalten. Dabei müssen die Tokens allerdings nicht zwingender maßen ganze Wörter sein (vgl. Manning, Raghavan & Schütze, 2008).

Tokenisierung wird oftmals als Auftrennung des Textes anhand von Leerzeichen verstanden. Dies ist allerdings nicht ganz korrekt. So kann es aus diversen Gründen nötig sein, eine durch Leerzeichen von anderen Zeichen getrennte Zeichenfolge noch weiter zu unterteilen, oder mehrere durch Leerzeichen voneinander getrennte Zeichenfolgen zu einen Token zusammenzufassen (z.B.: Zahlen wie 10 000). (vgl. Evert & Fitschen, 2001)

Mit folgenden Beispiel (vgl. Manning et al., 2008) werden kurz die Schwierigkeiten und Probleme bei der Tokenisierung veranschaulicht, die Verdeutlichen, dass bereits der erste Vorverarbeitungsschritt bei der Textverarbeitung eine Herausforderung darstellt.

Folgender Satz soll in Tokens aufgeteilt werden:

Mr. O'Neill thinks that the boys' stories about Chile's capital aren't amusing.

Der einfachste Weg diesen Satz in Tokens zu unterteilen ist es, in anhand von nicht alphanumerischen Zeichen aufzutrennen. Das Ergebnis würde dann Folgendermaßen aussehen (die Einzelnen Tokens werden durch eckige Klammern abgegrenzt):

[Mr] [.] [O] ['] [Neill] [thinks] [that] [the] [boys] [stories] [about] [Chile] ['] [s] [capital] [aren] ['] [t] [amusing] [.]

Wie man erkennt kann gibt es in diesem Satz einige Tokens bzw. Wörter, die zu Problemen führen können. So gibt es beispielsweise viele Möglichkeiten, wie man den Namen *O'Neill* alternativ in Tokens unterteilen kann:

- [O'] [Neill]
- [O'Neill]
- [O] [Neill]

Das Hauptproblem dabei ist nun, welche der vielen Möglichkeiten im Endeffekt die Sinnvollste bzw. die Wünschenswerteste ist. Selbst wenn von mehreren möglichen Alternativen eine ausgewählt wird, so ist relativ unwahrscheinlich, dass diese Aufteilung auch für andere ähnliche Zeichenfolgen zielführend ist. Dementsprechend stellt es eine große Herausforderung dar, für solche Zeichenfolgen allgemein gültige Regeln aufzustellen (vgl. Manning et al., 2008).

Um eine zuverlässige Tokenisierung durchführen zu können, schlagen Evert und Fitschen (2001) nachfolgenden Algorithmus vor. Als ersten Schritt trennt man alle Satzzeichen und sonstige nicht alphanumerische Zeichen von den Ziffern und Buchstaben. Im zweiten Schritt werden nun einige dieser Tokens mit Hilfe von regulären Ausdrücken zu einem einzigen größeren Token zusammengefügt. Dies soll verhindern, dass beispielweise Zahlen wie *10 000* oder Namen wie *O'Neill* aufgetrennt werden. Dies bedeutet allerdings das Entstehen einer Abhängigkeit der Tokenisierung von den regulären Ausdrücken. Je umfangreicher und zuverlässiger diese regulären Ausdrücke sind, desto zufriedenstellender wird das Ergebnis ausfallen. Das Problem hierbei ist jedoch, dass eine komplette Beschreibung einer Sprache mit solchen regulären Ausdrücken nahezu unmöglich ist. Darüber hinaus ist Wahrscheinlichkeit, dass das Ergebnis keine Fehler aufweist relativ gering (vgl. Evert & Fitschen, 2001).

Ein der Tokenisierung relativ ähnlicher Schritt, welcher bei der Textvorverarbeitung in den meisten Fällen ebenfalls notwendig erscheint, ist die Satzgrenzenerkennung, welche es ermöglicht, aus einem Text die einzelnen Sätze extrahieren zu können. Die Satzgrenzenerkennung ist in vielen Fällen trivial, da dabei prinzipiell nur die Zeichen, die ein Satzende im Text darstellen, identifiziert werden müssen, damit der Text anschließend dementsprechend aufgeteilt werden kann. Das größte Problem hierbei ist der Punkt als Satzende. Es ist natürlich möglich, dass ein Punkt auch eine andre Bedeutung, neben der als Satzzeichen, aufweist. So kann ein Punkt beispielsweise ein Teil einer Abkürzung sein und sollte deshalb in diesem Falle nicht als Satzende interpretiert werden. Auch hierbei gibt es mehrere Möglichkeiten diese beiden Fälle voneinander zu trennen. Einerseits ist es möglich mit Heuristiken ein Satzende zu erkennen. Eine Solche Heuristik ist zum Beispiel, dass, wenn ein Punkt ein Satzende markiert, das Nachfolgende Token mit relativ großer Wahrscheinlichkeit mit einem Großbuchstaben beginnt. Anderseits gibt es die Möglichkeit mit statistischen Methoden, indem man alle Tokens mit einem Punkt am Ende untersucht. Tokens die in einem Text ausschließlich mit einem Punkt am Ende vorkommen, sind mit großer Wahrscheinlichkeit eine Abkürzung. Allerdings gibt es auch bei diesen Verfahren immer wieder Unsicherheiten bzw. falsche Ergebnisse (vgl. Evert & Fitschen, 2001).

2.2.1.2 Wortartenerkennung

Die Wortartenerkennung (engl. part of speech - tagging), auch Wortartendisambiguierung genannt, dient dazu, zu jedem Wort dessen richtige Wortart, wie beispielweise Substantiv, Verb, Adjektiv etc. (vgl. 2.1.2.4), zu ermitteln. Dabei wird dem Wort ein sogenannter Tag aus einer vorher definierten und begrenzten Menge von Tags, dem sogenannten Tagset, zugewiesen. Dieses Tagset muss nicht zwangsweise nur aus den einzelnen Wortarten bestehen, oftmals werden auch Unterscheidungen innerhalb von Wortarten getroffen, um eine bessere Zuweisung treffen zu können (vgl. Manning & Schütze, 1999).

Bei der Wortartenerkennung ergibt sich auch eine Abhängigkeit von der vorangehenden Aufspaltung von Texten in Token. Dies lässt sich am besten anhand eines Beispiels mit der Adverbialphrase *des Weiteren* zeigen. Wenn diese Phrase in zwei Tokens ([des] und [Weiteren]) aufgeteilt wird, so kann es sein, dass die Adverbialphrase nicht mehr korrekt erkannt wird, da [des] als Artikel und [Weiteren] als Adjektive identifiziert wird. Solche Mehrdeutigkeiten, entsprechend dem Kontext, korrekt aufzulösen, ist nach Jurafsky und Martin (2008) die Hauptaufgabe von Taggern (vgl. Evert & Fitschen, 2001).

Um solche Ambiguitäten auflösen zu können, gibt es nun 2 Arten von Taggern. Einerseits gibt es den regelbasierten Tagger, andererseits den stochastischen Tagger. Der regelbasierte Tagger arbeitet mit vorher definierten und erarbeiteten Listen, welche Regeln beinhalten, die es ermöglichen, ein Wort mit einem bestimmten Tag zu versehen. Diese Regeln müssen für ein zufriedenstellendes Ergebnis sehr umfangreich sein und können nicht automatische erstellt werden. Stochastische Tagger verwenden eines Trainingskorpus und versuchen damit, die Wahrscheinlichkeit zu bestimmen mit der ein bestimmter Tag zu einem bestimmten Wort in einem bestimmten Kontext zugeordnet werden kann(vgl. Evert & Fitschen, 2001).

Generell gilt, dass gute Tagger in Abhängigkeit von der Textsorte zwischen 90 bis 97 % der Wörter richtig beurteilen. Im Normalfall haben stochastische Tagger eine geringere Fehlerquote als regelbasierte Tagger, allerdings haben beide sowohl Vorteile als auch Nachteile. Die Qualität des Taggers hängt natürlich von der Größe des Tagsets ab sowie von der Größe und Qualität des Trainingskorpus bzw. der Vollständigkeit der Regeln. Im Allgemeinen werden die Ergebnisse schlechter, je größer und damit je spezifischer das Tagset ist, um so besser, je vollständiger und qualitativ besser die Regeln beziehungsweise die Trainingskorpera sind (vgl. Jurafsky & Martin, 2008; Evert & Fitschen, 2001).

2.2.1.3 Lemmatisierung und Stemming

In einem Text kommen, aufgrund von grammatikalischen Regeln, viele verschieden Flexionsformen eines Wortes vor. Um diese Wörter dann in weiterer Folge als ein

und dasselbe Wort erkennen zu können, ist es nötig, die verschiedensten Flexionsformen der Wörter auf ihre Grundform zurückzuführen. Für eine Grundformreduktion gibt es zwei Arten. Einerseits gibt es die Lemmatisierung, andererseits das Stemming. Beide Methoden haben prinzipiell das gleiche Ziel, unterscheiden sich aber dennoch durch die Vorgehensweise. Bei der Lemmatisierung wird mit Hilfe eines umfangreichen Vokabulars und mit morphologischen Analysen versucht, die Flexionsendungen zu entfernen und das sogenannte Lemma, also die Grundform zurückzuliefern. Diese Grundform ist meist jene Form, wie sie auch in Wörterbüchern und Nachschlagewörtern vorkommt. Im Gegensatz dazu, versucht das Stemming mit Hilfe von Heuristiken die Flexionsendungen abzuschneiden, um so auf eine Grundform zu kommen (vgl. Manning et al., 2008).

Beide Methoden haben unterschiedliche Stärken und Schwächen. Während Stemming üblicherweise Probleme mit nicht regelmäßig konjugierbaren Wörtern hat, so ist die Lemmatisierung oftmals mit den unterschiedlichsten Flexionsformen überfordert. Ein großer Vorteil von Stemmern ist, dass er im Vergleich zu einem Lemmatisierer weniger Information als Ausgangspunkt benötigt. Darüber hinaus arbeitet ein Stemmer wesentlich schneller. Nachteilig zu erwähnen ist allerdings, dass der Stemmer keine Wörterbuchform als Grundform zurückliefert. Dies ist allerdings in den meisten Fällen vernachlässigbar, da es prinzipiell nur wichtig ist, dass der Algorithmus die unterschiedlichsten Flexionsformen eines Wortes auf ein und dieselbe Grundform zurückführt, damit diese ident erkannt werden (vgl. Manning et al., 2008).

Der am weitesten verbreitete Stemming Algorithmus ist der Porter Stemmer von M.F. Porter. Das Ziel das Porter mit seinem Algorithmus verfolgte war einerseits eine Methode anzubieten, die möglichst performant arbeitet, andererseits aber auch zufriedenstellende Ergebnisse liefert. Der Algorithmus von Porter verarbeitet in 5 Schritten mehr als 60 verschiedene Verkürzungsregeln, die nach bestimmten Kriterien angewandt werden, ab. Diese Kriterien sind bestimmte Vokal/Konsonanten - Folgen, die, falls diese zutreffend sind, eine Abtrennung eines Suffixes zur Folge haben (vgl. Porter, 1997).

2.2.1.4 Eigennamenerkennung und Chunking

Eigennamenerkennung und Chunking sind Vorverarbeitungsschritte, die sich sehr ähnlich sind. Prinzipiell geht es bei beiden darum, spezielle Wörter in Texten zu finden und diese zu markieren. Die Eigennamenerkennung dient dazu, Namen von Personen, Orten und Organisationen zu identifizieren. Um dies zu bewerkstelligen werden meistens vorgefertigte Listen verwendet, die dann mit dem Text abgeglichen werden (vgl. Stock, 2007).

Die Grundidee beim Chunking ist, spezielle Strukturen in Texten aufzuspüren und diese zu markieren. Solche Strukturen sind beispielsweise Nominalgruppen (z.B.: der große, alte Mann) und Verbalgruppen (wird ausgewählt werden). Diese Methoden können zu erheblichen Verbesserungen der Ergebnisse im Bereich der maschinellen Verarbeitung natürlicher Sprache führen (vgl. Evert & Fitschen, 2001).

2.2.1.5 Koreferenz Auflösung

Die Koreferenzauflösung hat nach Mitkov (2005) in den letzten Jahren sehr viel an Bedeutung gewonnen und nimmt in vielen Bereichen der Sprachverarbeitung, wie beispielsweise bei der automatischen Übersetzung, Informationsextraktion, automatischer Textzusammenfassung usw., eine wichtige Rolle ein. Eine Koreferenz ist ein Wort, das als Platzhalter für ein anderes Wort oder einen anderen Begriff dient. Neumann (2001) nennt drei verschiedene Arten von Koreferenzen, die im Bereich der Sprachverarbeitung von Interesse sind:

1. *Eigennamenkoreferenz*: Die Ausdrücke Präsident Bush, George W. Bush und Bush beziehen sich auf die gleiche Person und können daher als inhaltlich ident angesehen werden.

2. *Referenzen zwischen Designatoren*: Die Begriffe das Unternehmen, die Firma, der Konzern aus Redmond usw. können sich in einem Text auf ein und dasselbe beziehen, in diesem Falle Microsoft.

3. *Pronominale Referenzen* sind Referenzen, bei denen sich Pronomen (er, sie, es etc.) auf andere Wörter beziehen. Auch diese Referenzen können prinzipiell als identisch mit dem bezeichneten Objekt gesehen werden.

Bereits eine relativ flache Koreferenzauflösung kann zu erheblichen Verbesserungen der Ergebnisse führen (vgl. Neumann, 2001).

2.2.2 Text Analyse

Aufbauend auf die in Kapitel 2.2.1 beschriebenen Vorverarbeitungsschritte werden in diesem Unterkapitel einige Analyseverfahren, welche bei der Extraktion von semantisch relevanten Inhalten und auf ähnlichen Forschungsgebieten Anwendung finden, aufgezeigt.

2.2.2.1 Worthäufigkeiten

Das Zählen der Häufigkeit des Auftretens eines Wortes ist die einfachste Möglichkeit der Textanalyse in Hinblick auf das Auffinden von semantisch wichtigen Inhalten. Der Autor Luhn erklärte bereits 1958 in seinem Artikel *The Automatic Creation Of Literature Abstracts* den Zusammenhang zwischen der Häufigkeit des

Auftretens eines Wortes und der semantischen Relevanz dieses Wortes in einem Text. Luhn erkannte in seinen Forschungen, dass die Wörter mit den größten Häufigkeiten und Wörter, die nur sehr selten vorkommen weniger signifikant für den Text sind als jene, deren Häufigkeiten irgendwo im mittleren Bereich liegen (vgl. Luhn, 1958).

Der Grund dafür liegt auf der Hand. So sind die am meisten verwendeten Wörter der englischen Sprache Folgende: *the*, *be*, *to*, *if*, *and*, *a*, und *in* (vgl. Oxford, 2010). Solche und ähnlich allgemeine Wörter werden Stoppwörter genannt. Diese müssen im Zuge der Analyse herausgefiltert werden, da Stoppwörter prinzipiell eine geringe Aussagekraft in Bezug auf den Inhalt eines Textes aufweisen. Um dies zu bewerkstelligen werden oftmals so genannte Stoppwortlisten verwendet, mit denen es möglich ist, die Stoppwörter durch Abgleich des Textes mit diesen Listen zu entfernen. Allerdings ist es oftmals relativ schwierig, Stoppwörter zu definieren. So ist das Wort bzw. der Buchstabe *a* in der englischen Sprache im Normalfall ein Artikel oder eine Präposition und weist damit keine signifikante, inhaltliche relevante Bedeutung auf. Allerdings kann dieses Wort in einem Text mit medizinischem Hintergrund von essentieller Aussagekraft sein, da es beispielsweise die Bezeichnung für das Vitamin A sein kann (vgl. Meadow, 1992).

Eine Voraussetzung für das Funktionieren dieser Analyse ist nach Luhn (1958) die Reduktion der Wörter auf ihre Grundform (vgl. Kapitel 2.2.1.3). Dies dient dazu, die verschiedensten Flexionsformen eines Wortes zu identifizieren, um diese dann gleichzusetzen zu können und um somit die tatsächliche Anzahl des Vorkommens dieses Wortes ermitteln zu können (vgl. Luhn, 1958).

Die Methode der Worthäufigkeit beruht auf zwei Annahmen (vgl. Luhn, 1958):

1. Die Wahrscheinlichkeit, dass ein und dasselbe Wort in einem Text zwei verschiedene Bedeutungen aufweist, ist relativ gering.

2. Autoren eines Textes verwenden als Stilmittel oftmals viele Synonyme für wichtige Begriffe. Allerdings nimmt Luhn an, dass sich sowohl die Synonyme, die für einen Begriff nur in begrenzter Anzahl zur Verfügung stehen, als auch der Begriff selbst wiederholen. Dadurch erreichen dieses Wörter auch entsprechend häufig vor, um dadurch als wichtig erkannt zu werden.

Allerdings berücksichtigt diese Form der Analyse keine logischen oder semantischen Beziehungen zwischen den einzelnen Worten und Sätzen. Ein weiterer großer Nachteil dieser Methode ist die Tatsache, dass es kaum möglich ist herauszufinden, in welchem Bereich der Worthäufigkeiten sich die relevanten Wörter befinden, da diese Häufigkeiten klarerweise dokumentabhängig sind. Mit der in diesem Abschnitt erläuterten Methode von Luhn ist die Wahrscheinlichkeit, eine

große Menge für den Inhalt signifikante Wörter herauszufiltern relativ groß, es ist allerdings sehr unwahrscheinlich, dass man aus dieser Menge jene auswählen kann, die tatsächlich den Inhalt am besten repräsentieren (vgl. Meadow, 1992).

2.2.2.2 Verwendung von Korpora

Eine deutliche Verbesserung der Ergebnisse aus der in Abschnitt 2.2.2.1 vorgestellten Methode kann man auch damit erreichen, die Funktionswörter aus der Menge der potentiellen Kandidaten zu entfernen und nur mehr Inhaltswörter zu verwenden. Inhaltswörter sind Wörter, die, wie der Name schon sagt, bedeutungstragend sind (vgl. 2.1.2.4). Eine Möglichkeit dies zu bewerkstelligen, bietet sich im Einsatz von Korpora.

Ein Korpus ist nach (McEnery & Wilson, 2005) nichts anderes als eine Sammlung von mindestens zwei Texten. Die dafür verwendeten Texte sollen für den Zweck bzw. die Kategorie des Korpus repräsentativ sein, damit zuverlässige Aussagen über diese Textart getroffen werden können. So gibt es zum Beispiel Korpora die eine ganze Sprache in ihrer Gesamtheit abdecken sollen, aber auch Korpora, die nur ein Teilgebiet, wie beispielsweise die Literatur in Österreich im 19. Jahrhundert, abzielen. Das bekannteste Korpus ist das *Brown Korpus,* der an der Brown Universität in den 60er und 70er Jahren entwickelt wurde und als ausbalanciertes Korpus das amerikanische Englisch repräsentieren soll (vgl. Manning & Schütze, 1999).

Um die Funktionswörter aus der Menge der relevanten Wörter zu entfernen, schlägt der Autor Stubbs (2002) vor, die 100 häufigsten Wörter eines repräsentativen Korpus nicht für die Ermittlung der Worthäufigkeit zu verwenden. Er verwendete für seine Versuche das LOB - Korpus (Lancaster - Oslo - Bergen - Korpus), welches das britische Gegenstück zum Brown Korpus darstellt. Seinen Untersuchungen zur Folge befinden sich dann in der gekürzten Liste vornehmlich Inhaltswörter, die noch dazu größtenteils Schlüsselwörter für den Text sind und den Inhalt und die zentralen Themen des analysierten Dokumentes repräsentieren (vgl. Stubbs , 2002).

Wie die Untersuchungen von Stubbs (2002) zeigen, ist es bereits mit relativ einfachen Mitteln möglich, zufriedenstellende Ergebnisse zu erreichen. Allerdings ist es nötig, die Cut-Off-Linie beim entfernen der Funktionswörter in einer geeigneten Art und Weise zu definieren, da es bei kürzeren Texten auch dazu kommen kann, dass keine 100 verschiedene Wörter, welche auch genügend oft im Text vorhanden sind, zur Verfügung stehen. Um die Ergebnisse weiter zu Verbessern und die Abhängigkeit dieser von der verwendeten Textart zu minimieren, ist es nötig, weitere Analyseverfahren der Worthäufigkeitsanalyse hinzuzufügen, die in den nachfolgenden Abschnitten näher erläutert werden (vgl. Stubbs , 2002).

2.2.2.3 Wortähnlichkeiten

Ein weiteres Problem bei der rein statistischen Analyse von Luhn (vgl. Abschnitt 2.2.2.1) bzw. auch beim korpusunterstützten Ansatz von Stubbs (vgl. Abschnitt 2.2.2.2) ist, dass hierbei semantische Ähnlichkeiten zwischen Wörtern nicht berücksichtigt werden. Ähnliche Probleme finden sich in vielen Bereichen des Natural Language Processing, wie beispielsweise auch beim Information Retrieval. Auch beim Information Retrieval ist es schwierig, Informationen zu finden, die zwar inhaltlich mit dem gesuchten Übereinstimmen, aber die Wörter der Suchanfrage nicht im gewünschten Dokument vorkommen, da für dieses Dokument andere bedeutungsähnliche oder bedeutungsgleiche Wörter, sogenannte Synonyme, verwendet werden. Um dieses Problem zu lösen, wird beim Information Retrieval versucht, diese semantischen Verbindungen zwischen diesen Wörtern zu finden, um sie dann in weiterer Folge bei den Berechnungen zu berücksichtigen. (vgl. Lehmnitzer & Zinsmeister, 2006). Dieses Konzept kann auch in dem Bereich der statistischen Analyseverfahren eingesetzt werden, indem man die semantischen Ähnlichkeiten von verschiedenen Wörtern in einem Dokument ermittelt und diese dann bei der Berechnung der Häufigkeiten berücksichtigt.

Eine Möglichkeit der Ermittlung semantischer Relationen zwischen einzelnen Wörtern bietet sich in lexikalisch semantischen Wortnetzen. Eines der bekanntesten Wortnetze ist das Princeton WordNet (vgl. Kapitel 6.2). Dieses und ähnliche Wortnetze beinhalten einen Großteil der Wörter einer Sprache und stellen darüber hinaus auch Querbeziehungen diesen Wörtern her. In solchen Wortnetzen ist ein Wort ein Knoten, der mit vielen anderen Wörtern, verbunden ist. Die Verbindungen dieser Konten repräsentieren dabei semantische Relationen. Je kleiner die Anzahl der Knoten zwischen 2 Wörtern ist, desto größer die semantische Ähnlichkeit zwischen zwei Wörtern. Des Weiteren ist es mit solchen Wortnetzen unter anderem auch möglich, spezielle Wörter, die mit einem bestimmten Wort in Verbindung stehen, wie beispielsweise Hyperonyme, Hyponyme und Synonyme zu extrahieren. Hyperonyme sind Oberbegriffe, also eine Verallgemeinerung, Hyponyme Unterbegriffe und Synonyme sind bedeutungsgleiche oder bedeutungsähnliche Begriffe (vgl. Stock, 2007; Miller, 1999).

2.2.2.4 Weitere Textanalyseverfahren

Die in den vorhergehenden Abschnitten vorgestellten Textanalyseverfahren dienen dazu, den von Luhn (1958) entwickelten Ansatz der Extraktion von Sätzen zur Automatischen Textzusammenfassung zu unterstützen und weiterzuentwickeln, mit dem Ziel, die semantisch relevanten Inhalte aus den Texten identifizieren zu können. Es gibt noch sehr viele weitere Verfahren, um die Ergebnisse der Textanalyse zu verfeinern. Einige dieser Verfahren werden in den nächsten beiden Kapiteln vorgestellt.

All diese Ansätze, wie zum Beispiel auch die Wortbedeutungsunterscheidung (vgl. Kunze, 2001), das mit einbeziehen von Chunks und Eigennamen (vgl. 2.2.1.4), die Verwendung der Koreferenzauflösung (vgl. 2.2.1.5) usw. ergeben erst dann einen Sinn, wenn ein geeigneter Ansatz gefunden wird, der die einzelnen Analyseverfahren miteinander kombiniert. Ein dafür geeigneter Ansatz ist die einzelnen Wörter in Abhängigkeit dieser Analyseverfahren entsprechend zu gewichten und danach die wichtigsten Wörter und Phrasen anhand dieser Gewichte aus dem Text zu extrahieren. Diese Methode wird in den folgenden Abschnitten aufgegriffen und daher wird in diesem Kapitel nicht näher darauf eingegangen.

2.3 Zusammenfassung

In diesem Kapitel wurde näher auf die Grundlagen und Hintergründe eingegangen, welche für eine Extraktion von semantisch relevanten Daten aus natürlich sprachlichen Inhalten von Bedeutung sind. Zu Beginn wurden wichtige Begriffe der Sprachwissenschaft und der Linguistik definiert und danach wurde näher auf das Gebiet der maschinellen Verarbeitung natürlicher Sprache eingegangen. Dabei wurde versucht, aus diesem sehr umfangreichen Gebiet die wichtigsten Themen, die in Zusammenhang mit dem Thema dieser Arbeit stehen, zu extrahieren. Der Fokus wurde dementsprechend vor allem auf maschinelle Verarbeitung von Texten, in Hinblick auf eine semantische Analyse dieser, gelegt. Dabei zeigte sich bereits die Komplexität der natürlichen Sprache einhergehend damit, dass die Extraktion von inhaltlich relevanten Daten keineswegs eine Trivialität darstellt sondern vielmehr eine große Herausforderung ist.

Trotz dieser Komplexität konnten aus diesem Kapitel dennoch viele wichtige Erkenntnisse gewonnen werden. So zeigte sich ein Zusammenhang zwischen bestimmten Wortgruppen und der Relevanz dieser Wörter bezüglich des Textes. Diese Wortgruppen sind vorzugsweise Substantive obgleich auch andere Wortarten (prinzipiell alle Inhaltswörter) durchaus bedeutungstragend sein können. Aus diesem Grund ist es im Normalfall auch nötig, die Funktionswörter bzw. Stoppwörter zu ignorieren. Dies ist vor allem bei der statistischen Analyse unabdinglich, da bei dieser Methode die Häufigkeit des Wortes im Text Aufschluss über die Wichtigkeit des Wortes für den Inhalt gibt. Werden die Stoppwörter nämlich nicht entfernt, sind die inhaltsrelevanten Wörter nicht mehr durch ihre Häufigkeit zu identifizieren, da Stoppwörter in Texten sehr häufig vorkommen.

Des Weiteren zeigte sich, dass für eine erfolgreiche Textanalyse gewisse Vorverarbeitungsschritte benötigt werden, um aus dem Text, die gewünschten Informationen extrahieren zu können. Solche Methoden sind beispielsweise Tokenisierung, Wortartenerkennung, Satzgrenzenerkennung, Grundformreduktion, Koreferenzauflösung und auch Eigennamenerkennung. Darüber hinaus zeigt dieses

Kapitel, dass die Ergebnisse der Analysen sehr von der Qualität dieser Vorverarbeitungsschritte abhängig sind.

Dieses Kapitel veranschaulichte ebenfalls, dass eine Extraktion von semantisch relevanten Daten nur dann sinnvoll und möglich ist, wenn auch semantische Analysen des Textes durchgeführt werden. Dafür werden oftmals semantische Netzwerke verwendet. Bei genauerer Betrachtung von semantischen Analyseverfahren wird allerdings deutlich, dass eine domänenunabhängige semantische Analyse sehr schwer durchführbar ist und deshalb die Verwendung von spezifischen Korpora durchwegs Vorteile mit sich bringt.

Die Grundlage für die Extraktion von semantisch relevanten Daten aus natürlich sprachlichen Inhalten bildet das Gebiet der maschinellen Verarbeitung von natürlicher Sprache. Viele Teilbereiche dieser wissenschaftlichen Domäne sind dem Thema dieser Arbeit sehr ähnlich bzw. bieten weitere Konzepte und Ideen, die für eine Extraktion von semantisch relevanten Daten von Bedeutung sind. Da dieses wissenschaftliche Teilgebiet sehr umfangreich ist, werden im nachfolgenden Kapitel jene Bereiche des NLP vorgestellt, die einerseits für das weitere Verständnis dieser Arbeit erforderlich sind und andererseits dem Zweck dieser Arbeit dienlich sind.

3 Anwendungsgebiete des Natural Language Processing

Die Anwendungsgebiete der maschinellen Verarbeitung natürlicher Sprache sind sehr vielfältig und zahlreich. Carstensen (2001) unterteilt die Anwendungen nach Funktionalität und nennt Folgende als Beispiele: Rechtschreibkorrektur, Volltextsuche und Textmining, computerunterstützte Lexikologie, maschinelle Übersetzung, Textklassifikation, Informationsextraktion, Textzusammenfassung, Sprachsynthese- und Spracherkennungssysteme, Information Retrieval - Systeme, Sprachlehr- und Sprachlernsysteme sowie einige weitere. Auf all diese Bereiche näher einzugehen würde den Rahmen dieser Arbeit sprengen und ist daher nicht zielführend. Aus diesem Grunde werden nachfolgend stellvertretend diejenigen Anwendungsgebiete herausgefiltert und näher erläutert, die sehr eng mit der Extraktion von semantisch relevanten Daten aus natürlich sprachlichen Inhalten verflochten sind und darüber hinaus wichtige Erkenntnisse liefern.

3.1 Automatische Übersetzung

Der automatischen maschinellen Übersetzung kommt in der heutigen globalisierten Welt, in der die problemlose Kommunikation von Menschen mit verschiedensten sprachlichen Hintergründen ein zentrales Thema darstellt, eine immense Bedeutung zu. Ein zentraler Aspekt der der automatischen Übersetzung ist es die Bedeutung und den Kontext des zu übersetzenden Textes richtig zu deuten und die Begriffe dahingehend zu übersetzen. Dadurch begründet sich auch die enge Verknüpfung der maschinellen Übersetzung mit der Extraktion von semantisch relevanten Daten (vgl. Dorna, 2001).

Nach Hutchinson und Sommers (1992) gibt es mehrere Möglichkeiten, die maschinelle Übersetzung zu unterteilen. So gibt es bilinguale, also System, die nur zwischen 2 Sprachen übersetzen, und multilinguale Systeme, die zwischen mehreren Sprachen Übersetzen. Die bilingualen Systeme können darüber hinaus sowohl unidirektional als auch bidirektional sein. Eine weitere Möglichkeit der Unterteilung bietet sich anhand der Übersetzungsstrategien: es gibt die direkte Übersetzung, Transfersysteme und Interlinguas (vgl. Hutchinson & Sommers, 1992).

Die direkte Übersetzung ist die einfachste Übersetzungsmethode und verwendet bilinguale Wörterbücher, um den zu übersetzenden Text direkt zu einem Text einer anderen Sprache zu transformieren. Dabei wird weder eine syntaktische, noch eine semantische Analyse des Eingangstextes bzw. des Ausgangstextes durchgeführt. Die Methode der direkten Übersetzung liefert zwar kaum syntaktisch oder semantisch korrekte Übersetzungen, die großen Vorteile dieser Methode sind

aber die Robustheit und die Geschwindigkeit (vgl. Dorna, 2001; Hutchinson & Sommers, 1992).

Bei der Transfermethode wird der Eingangstext analysiert und in eine spezielle abstrakte Repräsentation umgewandelt. Diese Repräsentation ist sprachspezifisch und nicht bidirektional, d. h. es gibt beispielsweise für die Übersetzung vom Deutschen ins Englische eine andere Repräsentation als für die Übersetzung vom Englischen in die deutsche Sprache. Die Generierung des Ausgangstextes erfolgt danach anhand von bestimmten Regeln, die diese abstrakte Repräsentation in den Ausgangstext umwandeln (vgl. Dorna, 2001; Hutchinson & Sommers, 1992).

Die Interlinguamethode verwendet ebenfalls eine abstrakte Zwischenrepräsentation, welche allerdings bei dieser Methode sprachunabhängig ist. Diese Repräsentation hat keinen direkten Bezug zu einer spezifischen Sprache und ist somit für die Verwendung in multilingualen Systemen von großer Bedeutung, da beim Hinzufügen einer neuen Sprache in das Übersetzungssystem nur mehr Module erzeugt werden müssen, welche die Sprache in die abstrakte Zwischenrepräsentation umwandeln bzw. vice versa. Das Problem bei diesem Ansatz ist, dass dieser ein vollständiges Verstehen der Sprachen voraussetzt. Darüber hinaus zeigte sich, dass die Erzeugung der sprachunabhängigen Zwischenrepräsentationen als äußerst komplex und nahezu unmöglich ist. (vgl. Dorna, 2001; Hutchinson & Sommers, 1992).

Alle der oben genannten Strategien bedingen für ein zufriedenstellendes Ergebnis ein semantisches, kontextabhängiges Verstehen des Ausgangstextes. Um diese zu bewerkstelligen gibt es nun laut Dorna (2001) drei verschiedene Konzepte, die in ähnlicher Art und Weise auch bei der Extraktion von semantisch relevanten Daten eingesetzt werden können. Der wissensbasierte Ansatz versucht, die Umwelt domänenspezifisch zu modellieren, um so Mehrdeutigkeiten zwischen den Wörtern aufzulösen. Des Weiteren gibt es den beispielbasierten Ansatz, bei dem versucht wird, mittels eines möglichst großen bilingualen Korpus die richtige Übersetzung zu finden. Dieses Korpus kann sowohl Phrasen als auch ganze Sätze enthalten. Der dritte Ansatz ist die statistische Übersetzung. Dabei wird versucht, die wahrscheinlichste Übersetzung aus einer Menge von verschiedenen möglichen Übersetzungen auszuwählen (vgl. Dorna, 2001).

Der Autor Lehrberger (2003) greift im Zuge der automatischen Übersetzung den Ansatz der *Sublanguage* auf. Das Konzept der Sublanguage ist vergleichbar mit der in Abschnitt 2.2.2.2 beschriebenen Verwendung von Korpora. Sublanguage ist nichts anderes als ein kontextabhängiges Korpus. Dieses Korpus wird in Abhängigkeit der in einem speziellen Fachgebiet, wie beispielsweise Informatik, Mathematik, Biologie usw., vorkommenden Wörter und dazugehörigen Bedeutungen in diesem Gebiet, erstellt. Der große Vorteil dieses Konzeptes bietet

sich darin, dass dadurch die Wörter in einem spezifischen Kontext mit großer Wahrscheinlichkeit nur mehr eine Bedeutung haben (vgl. Lehrberger, 2003).

Der Nachteil dieser Konzepte ist, dass für diese Ansätze eine Textklassifizierung (vgl. Abschnitt 3.2) benötigt wird, um das geeignete Vokabular auswählen zu können.

3.2 Automatische Textklassifizierung

Die Aufgabe der automatischen Textklassifizierung, auch Textkategorisierung genannt, ist es, Dokumenten eine Kategorie aus einer vorher definierten Menge von Kategorien zuzuordnen. Solche Kategorien sind beispielsweise Medizin, Sport, Politik usw. (vgl. Sebastiani, 2002).

Der Autor Brückner (2001) unterteilt die automatische Textkategorisierung in die regelbasierten und in die statistischen Verfahren. Bei den regelbasierten Verfahren wird anhand von Regeln, oftmals boolesche Regeln, ein Text einer spezifischen Kategorie zugeordnet. Dies geschieht nur, wenn der Text die Regeln vollständig erfüllt. Diese Vorgehensweise ermöglicht auch Mehrfachklassifikationen, da ein Text Regeln für verschiedene Textkategorien erfüllen kann. Dieser Umstand kann natürlich von erheblichem Nachteil sein, da bei dieser Bewertung der Texte keine Unterscheidung getroffen werden kann, welche der ausgewählten Kategorien sich am besten dafür eignet (vgl. Brückner, 2001)

Für die statistischen Verfahren gibt es laut Brückner (2001) viele verschiedene Algorithmen, die im Bereich der automatische Textklassifikation in Verwendung sind:

1. *Rocchio - Algorithmus*: Der Rocchio Algorithmus verwendet das Vektorraummodell, um eine Menge von Texten zu klassifizieren. Dabei wird für jede Klasse ein sogenannter Zentroid – Vektor, der das Zentrum einer Klasse repräsentiert, gebildet. Ein neuer Text, repräsentiert durch einen Vektor, wird nun in jene Kategorie eingeordnet, zu deren Zentroid - Vektor dieser die geringste Distanz aufweist (vgl. Manning et al., 2008).

2. *k-Nearest – Neighbour – Algorithmus*: Bei diesem Algorithmus werden die Klassenzugehörigkeiten der k nächsten Nachbarn des einzuordnenden Dokumentes untersucht. Das Dokument wird dann jener Klasse zugeordnet, zu der die größte Anzahl dieser Nachbarn gehören (vgl. Manning et al., 2008).

3. *Support – Vector – Machine*: Die Methode der Support Vector Machine (SVM) ist ein Verfahren zur Mustererkennung. Anhand von

Trainingsbeispielen wird eine Hyperebene berechnet, die die optimale Trennung der positiven und der negativen Trainingsbeispiele darstellt. Optimale Trennung bedeute dabei, dass jene Vektoren, welche die Texte repräsentieren und am nächsten zu dieser Ebene sind, einen maximalen Abstand zueinander haben. Diese Vektoren werden auch als Support Vectors bezeichnet. Die Zuordnung eines Textes wird anhand von Abstandsberechnungen zu den Support Vectors durchgeführt (vgl. Brückner, 2001; Sebastiani, 2002; Manning et al., 2008)

4. *Naive – Bayes – Klassifikators*: Bei dieser von Lewis (1992) erstmals veröffentlichte Methode der Textkategorisierung wird die Wahrscheinlichkeit, dass ein bestimmtes Dokument in eine bestimmte Klasse gehört, errechnet. Dies geschieht über die Berechnung der bedingten Wahrscheinlichkeiten $P(t_k|c)$ für jedes Wort, wobei t_k das k-te Wort eines Dokumentes und c die Klasse repräsentiert. $P(t_k|c)$ ist die Wahrscheinlichkeit, dass das Wort t_k in einem Dokument der Klasse c vorkommt. P(c) ist die A-Priori-Wahrscheinlichkeit dass ein Dokument zur Klasse zugeordnet werden kann. Mit Hilfe der Formel von Bayes, der bedingten Wahrscheinlichkeit $P(t_k|c)$ und der A-Priori-Wahrscheinlichkeit P(c) wird nun für jedes Dokument und für jede der Klassen die Wahrscheinlichkeit berechnet, mit der das Dokument einer Klasse zugeordnet werden kann. Das Dokument gehört dann zu jener Klasse, bei der die Wahrscheinlichkeit am größten ist (vgl. Manning et al., 2008; Lewis, 1992).

Eine Textklassifikation bringt auch bei der automatischen Extraktion von semantischen Inhalten einen wesentlichen Vorteil. Durch Kenntnis der Kategorie können, bei Vorhandensein eines repräsentativen Korpus bzw. bei Kenntnis der repräsentativen Wörter dieser Kategorie, viele Wörter und Phrasen ausgeschlossen werden, die für diese spezifische Kategorie nur wenig bis keine Aussagekraft haben. Dies kann zu erheblichen Verbesserungen der Ergebnisse führen.

3.3 Automatische Zusammenfassung

Die automatische Zusammenfassung ist mit der Extraktion von semantisch relevanten Daten aus natürlich sprachlichen Inhalten eng verflochten, da beide Teilgebiete der maschinellen Sprachverarbeitung versuchen, die wesentlichsten Inhalte aus einem Text zu extrahieren.

Die Autorin Sparck Jones (1999) unterscheidet bei der automatische Textzusammenfassung zwei verschiedene Ansätze: einerseits die *Textextraktion* und andererseits die Faktenextraktion. Bei der Textextraktion werden mit Hilfe von verschiedensten Methoden wichtige Segmente, meistens einzelne Sätze, aus dem

Ursprungstext extrahiert. Dabei findet aber keine tiefere semantische Analyse statt, sondern es werden oftmals nur rein statistische Methoden, wie beispielsweise Worthäufigkeiten, berücksichtigt. Darüber hinaus kommen oftmals sogenannte Cue – Phrase – Listen, welche spezielle Phrasen beinhalten, die auf wichtige Textpassagen hindeuten, zum Einsatz oder die Position des Satzes im Text beziehungsweise im Absatz wird berücksichtigt. Die Textextraktion ist rein extraktiv, da die Sätze, die ausgewählt werden, nicht verändert werden. Die bei der Textextraktion entstehenden Zusammenfassungen beinhalten dementsprechend eine bestimmte Anzahl von Einzelsätzen, die allerdings oftmals keinen direkten Zusammenhang untereinander haben (vgl. Sparck Jones, 1999).

Die Faktenextraktion ist eine Methode der Zusammenfassung bei der im ersten Schritt eine Zwischenrepräsentation der Daten erzeugt wird. Diese Zwischenrepräsentation soll dabei die wichtigsten semantischen Merkmale des Ausgangstextes repräsentieren. Diese Merkmale sind im Allgemeinen Wörter und Phrasen und können bei der Erzeugung dieser Repräsentation auch abgeändert werden, um den strukturellen Anforderungen dieser zu genügen. Aus dieser Zwischenrepräsentation wird dann anschließend ohne direkten Bezug zum Ausgangstext die natürlich sprachliche Zusammenfassung generiert (vgl. Sparck Jones, 1999).

Beide Methoden haben sowohl Vorteile als auch Nachteile. Die Textextraktion ist robuster und allgemeiner, liefert allerdings qualitativ schlechtere Ergebnisse. Diese Zusammenfassungen sind weder klar strukturiert noch zusammenhängend und sind somit auch keine wohl geformten, stilistisch akzeptierbare Texte. Dennoch können diese die essentiellen Inhalte eines Textes durchaus repräsentieren. Die Zusammenfassungen der Faktenextraktion hingegen liefern durchaus wohlgeformte Texte, haben allerdings den Nachteil, dass die Sichtweise der Zusammenfassungen sehr von den Formalismen abhängt, mit denen diese aus der Zwischenrepräsentation erzeugt werden. Dies kann dazu führen, dass die Sichtweise des Autors sich nicht unbedingt mit der Sichtweise der Zusammenfassung deckt (vgl. Sparck Jones, 1999).

3.4 Information Retrieval

Stock (2007) definiert Information Retrieval wie folgt: „Information Retrieval ist die Wissenschaft, die Technik und der Praxisbereich des Suchens und Findens von Informationen.“ (Seite 2). Der Focus des Information Retrieval liegt laut Baeza - Yates und Ribeiro - Neto (1999) auf dem Auffinden von Information und nicht auf dem Auffinden von Daten. Das Ziel des Information Retrieval ist es also, spezifische Information zu einem Thema zu finden und somit nicht alle Daten, die diesem Thema zuordenbar sind bzw. die der Suchanfrage genügen würden, darzulegen.

Hierbei zeigt sich bereits die Verbindung zum Thema dieser Arbeit. Auch beim Information Retrieval ist es wichtig, aus einer großen Menge von Daten, spezifische Daten, die wichtig erscheinen, herauszufiltern. Nach Grossman und Frieder (2004) gibt es nun verschiedenste Information Retrieval Strategien, die nachfolgend kurz erläutert werden.

3.4.1 Vektorraummodell

Beim Vektorraummodell (engl. Vector Space Model) werden sowohl die Menge der Dokumente als auch die Suchanfrage des Benutzers als Vektoren repräsentiert. Die einzelnen Terme dieser Vektoren sind gewichtete Indexterme, also die Wörter, die die einzelnen Dokumente bzw. die Suchanfrage beinhalten. Das Gewicht der Indexterme dient dazu, die Relevanz dieser Wörter für das Dokument zu beschreiben. Für die Gewichtung dieser Terme gibt es viele verschiedene Möglichkeiten, die einfachste davon ist die sogenannte Termfrequenz tf also die Anzahl der Vorkommnisse eines Terms in einem Dokument. Da allerdings ein Wort, welches in vielen Dokumenten vorkommt, wenig aussagekräftig ist, wird häufig die inverse Dokument Frequenz idf verwendet. Die idf ist ein Maß für die Relevanz des Wortes in allen Dokumenten. Das bedeutet, dass ein Wort das in vielen Dokumenten vorkommt, weniger relevant ist als ein Wort das in wenigen Dokumenten vorkommt (Baeza - Yates & Ribeiro - Neto, 1999).

Das Ziel des Vektorraummodells ist es ein Ähnlichkeitsmaß zwischen dem Suchvektor und dem Dokumentenvektor aufzustellen. Als Ähnlichkeitsmaß kann beispielsweise der Kosinus des Winkels zwischen diesen Vektoren dienen. Das Vektorraummodell hat den großen Vorteil, dass auch Dokumente als wichtig erkannt werden, selbst wenn diese die Suchanfrage nicht vollständig erfüllen. Der Nachteil dieser Methode ist allerdings, dass bei diesem Modell davon ausgegangen wird, dass die einzelnen Wörter unabhängig sind. Dies trifft aber im Allgemeinen nicht zu und kann somit die Performance des Algorithmus negativ beeinflussen (Baeza - Yates & Ribeiro - Neto, 1999).

Die Autoren Grossman und Frieder (2004) stellen eine weitere Möglichkeit vor, welche die Gewichtung der Terme weiter verfeinert und somit auch die Ergebnisse des Retrieval - Prozesses verbessert. Dieser Ansatz soll den Einfluss, den einzelne den sehr häufige Wörter haben, reduzieren, indem bei der Gewichtsberechnung nicht mehr die Termfrequenz, sondern der Logarithmus der Termfrequenz mit einbezogen wird. Des Weiteren zeigen Grossman und Frieder (2004) auch den Nachteil darin den Kosinus des Winkels der Vektoren als Ähnlichkeitsmaß zu nutzen und zeigen Alternativen, wie man diesen nachteiligen Effekt vermindern kann (vgl. Grossman & Frieder, 2004).

3.4.2 Probabilistic Model

Dieses Modell geht von der Annahme aus, das es für jede Suchanfrage einen bestimmte Menge von Dokumenten gibt, die ausschließlich jene Dokumente beinhaltet, die wirklich relevant sind und somit das ideale Ergebnis für diese Suchanfrage darstellen. Um nun diese Menge von Dokumenten bei einer Suchanfrage zu erhalten, ist eine exakte Beschreibung der idealen Antwort nötig. Diese ist allerdings zum Zeitpunkt der Suchanfrage nicht vorhanden und muss daher vorab geschätzt werden. Mit diesen geschätzten Werten ist es dann möglich, eine erste Vorauswahl an Dokumenten, die der Suchanfrage genügen, zu ermitteln. Diese Vorauswahl ist aufgrund der zuvor getroffen Annahmen nicht zwingendermaßen die ideale Auswahl, so dass es bei dieser Methode nötig ist, den Prozess iterativ zu wiederholen, um die aktuelle Auswahl schrittweise an die ideale Auswahl anzunähern. Dies geschieht mit Interaktion mit dem Benutzer, indem dieser die relevanten Dokumente aus der Vorauswahl selektiert, worauf dann die zuvor geschätzte Beschreibung der idealen Antwort adaptiert wird und so eine neue Menge an möglichen Dokumenten ausgewählt wird (Baeza - Yates & Ribeiro - Neto, 1999).

Das Probabilistic Model versucht über Wahrscheinlichkeitsberechnungen die Relevanz zu berechnen, mit der ein Dokument zur Suchanfrage passt. Wie schon beim Vektorraummodell gilt auch hier die Annahme, dass die einzelnen Wörter unabhängig voneinander sind. Die wesentlichsten Nachteile des Probabilistic Models sind einerseits, dass die Relevanz der Dokumente zu Beginn geschätzt werden muss und andererseits, dass die Häufigkeit des Vorkommens eines Wortes in einem Dokument nicht berücksichtigt wird, da die Gewichte bei dieser Methode binär sind (Baeza - Yates & Ribeiro - Neto , 1999).

3.4.3 Sprachmodelle

Der Ansatz der Sprachmodelle (engl. Language Models) ist konträr zu den in den Abschnitten 3.4.1 und 3.4.2 vorgestellten Algorithmen. Bei dieser Methode wird für jedes Dokument ein sogenanntes Sprachmodell erzeugt. Der Begriff Sprachmodell bezieht sich auf eine Wahrscheinlichkeitsverteilung basierende Textmodellierung. Anhand dieses Modells werden dann aus dem Ausgangsdokument bestimmte Wörter extrahiert und damit eine repräsentative Suchanfrage für das Dokument gebildet. Je ähnlicher diese Suchanfrage der Suchanfrage des Benutzers ist, desto größer die Relevanz des Dokuments (vgl. Zhai, 2008; Manning et al., 2008).

Das Hauptproblem beim Information Retrieval mit Sprachmodellen ist, ähnlich wie beim Probabilistic Model (siehe 3.4.2), das Abschätzen bzw. das Erzeugen des Sprachmodells. Die Ergebnisse die mit dieser Methode erzielt werden sind im Vergleich mit anderen probabilistischen Ansätzen durchaus akzeptabel, obgleich

dieses Methode von der Annahme bedingt, dass die Dokumente und die Suchanfragen sich eine Kategorie teilen (Manning et al., 2008).

3.4.4 Inferenz Modell

Das grundlegende Inferenz Modell des Information Retrieval besteht aus einem Dokumentnetzwerk und einem Suchanfragenetzwerk (vgl. Abbildung 1). Das Dokumentnetzwerk wird einmal für alle Dokumente, welche betrachtet werden, gebildet und repräsentiert diese. Das Suchanfragenetzwerk besteht aus einem einzelnen Knoten und bildet den Informationsbedarf einer Suchanfrage ab. Diese Repräsentation kann, wenn die Suchanfrage abgeändert wird, im Gegensatz zur Repräsentation eines Dokuments während des Informationsfindungsprozesses adaptiert werden, um den geänderten Informationsbedürfnissen gerecht zu werden. Alle Knoten des Dokumentnetzwerkes und des Suchanfragenetzwerkes sind untereinander verbunden und weisen als Inhalt ausschließlich binäre Werte auf (vgl. Turtle & Croft, 1991).

Ein Dokumentnetzwerk besteht aus Dokumentknoten (d_i), die Information darüber beinhalten, ob ein Dokument in Betracht gezogen wird, aus Textrepräsentationsknoten (t_j), welche Information enthalten, ob ein Text in Betracht gezogen wird und aus Konzeptrepräsentationsknoten(r_k). Die Wurzeln jedes Dokumentnetzwerkes sind die Dokumentknoten, die Textrepräsentationsknoten sind die inneren Knoten und die Konzeptrepräsentationsknoten sind die Blätter. Die jeweiligen Dokumentknoten sind genau mit einem Textrepräsentationsknoten verbunden wohingegen die Textrepräsentationsknoten beliebig viele Verbindungen zu Konzeptrepräsentationsknoten aufweisen können. Während des Bildens des Netzwerkes wird jedem Dokumentknoten die Wahrscheinlichkeit zugewiesen, mit der dieses Dokument in Betracht gezogen wird. Zu Beginn ist diese Wahrscheinlichkeit gleich dem Kehrwert der Anzahl der Dokumente. Jeder Textrepräsentationsknoten wird in Abhängigkeit seines Dokumentknotens und der Beziehung zu diesem bewertet. Wenn die Beziehung vollständig ist und der Status des Dokumentknotens true, also in Betracht gezogen, ist, dann ist auch der Status des Textrepräsentationsknoten true (vgl. Turtle & Croft, 1991).

Das Suchanfragenetzwerk hat ein einzelnes Blatt (I), gegebenenfalls mehrere Suchanfrageknoten(q_l), welche die Suchanfragen repräsentieren und Knoten, die die Konzepte des Informationsbedürfnisses(c_m) dieser Suchanfrage darstellen (vgl. Turtle & Croft, 1991).

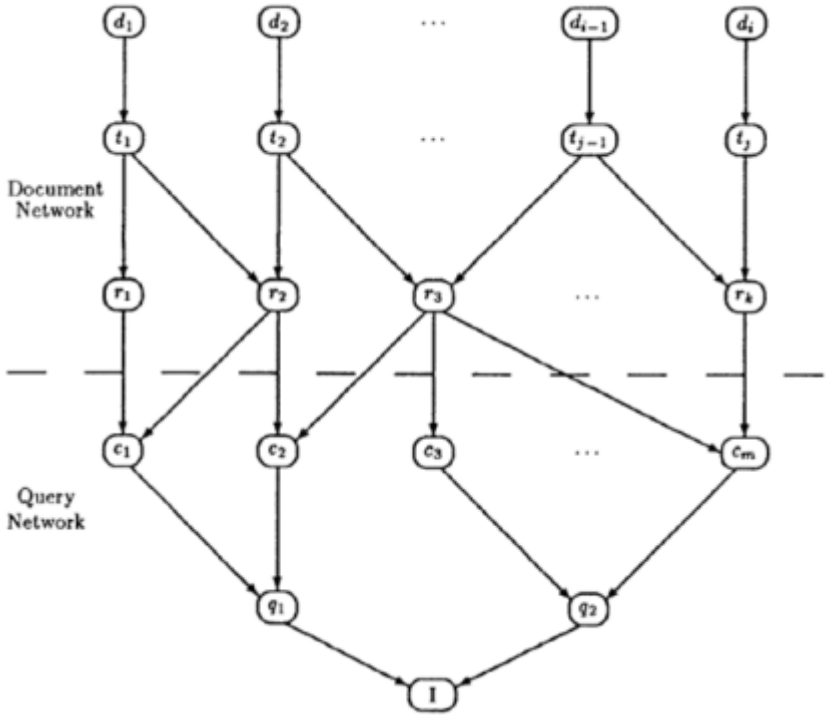

Abbildung 1: Inferenz Modell (vgl. Turtle & Croft, 1991).

Bei diesem Inferenz Modell wird angenommen, dass das Informationsbedürfnis einer Suchanfrage vollständig durch bestimmte Konzepte, wie beispielsweise Schlüsselwörter und Phrasen oder Beispieldokumentsammlungen, modelliert werden kann (vgl. Turtle & Croft, 1991).

Die Konzepte (c_m) stellen die Verbindung mit den Konzepten des Konzeptrepräsentationsknoten des Dokuments her und sind im besten Falle identisch. Mit Hilfe der Anfangswahrscheinlichkeiten und den Zuständen der Dokument- bzw. der Textrepräsentationsknoten kann nun jene Teilmenge aus der Menge der Dokumente ermittelt werden, die am besten zu der Suchanfrage passt. Dabei es ist auch möglich, die ermittelten Dokumente nach Relevanz zu ordnen (vgl. Turtle & Croft, 1991).

Eine vereinfachte Darstellung dieses Prinzips zeigen die Autoren Manning, et al. (2008). Sie reduzieren dieses Prinzip in ihren Erläuterungen auf die Dokumentknoten, auf Indextermknoten, welche den Textrepräsentationsknoten entsprechen und Suchanfrageknoten (Konzepte des Informationsbedürfnisses (c_m)). Die Relevanz des Dokuments wird bei diesem Modell berechnet, indem die Wahrscheinlichkeit in Abhängigkeit der Verbindungen zwischen den Indextermen und den Suchfrageknoten, sowie in Abhängigkeit der binären Werte dieser ausgewertet wird (Manning et al., 2008).

3.4.5 Boolesches Modell

Bei diesem Modell sind die Suchanfragen boolesche Ausdrücke, das bedeutet, die einzelnen Indexterme, welche zudem binär gewichtet werden, sind durch boolesche Operatoren miteinander verbunden. Diese Suchanfrage muss in disjunktiver Normalform vorliegen, damit die Relevanz eines Dokuments berechnet werden kann. Die Ähnlichkeit eines Dokuments d_m mit der Suchanfrage q ist definiert als:

$$sim(d_m, q) = \begin{cases} 1 \;\; falls \; \exists \overline{q_{cc}} \mid (\overline{q_{cc}} \; \epsilon \; \overline{q_{dnf}}) \wedge (\forall t_i, \; w_i(\overline{d_m}) = \; w_i(\overline{q_{cc}})) \\ 0 \;\; andernfalls \end{cases} \qquad 3.1$$

Der Ausdruck q_{cc} in Formel 3.1 entspricht den konjunktiven Komponenten der disjunktiven Normalform. Der Term t_i entspricht dabei dem i-ten Term der Suchanfrage und der Term $w_i(\overline{d_m})$ entspricht dem Gewicht des Wortes w_i aus dem Dokument d_m. Wenn diese Funktion 1 zurückliefert, dann ist das Dokument relevant, wenn das Ergebnis 0 ist, dann ist das Dokument nicht relevant (Baeza - Yates & Ribeiro - Neto, 1999).

Das Boolesche Modell trennt also die Dokumente in die Menge der relevanten und der nicht relevanten Dokumente auf. Bei diesem Modell ist es daher ohne zusätzliche Berechnungen nicht möglich, Reihungen der Dokumente vorzunehmen. Darüber hinaus ist es nicht möglich, Dokumente zu erhalten, die nur teilweise der Suchanfrage entsprechen. Für dieses Modell spricht allerdings, dass es relativ einfach und intuitiv ist. Darüber hinaus ist die Formulierung des Modells klar und eindeutig (Baeza - Yates & Ribeiro - Neto, 1999).

3.4.6 Latent Semantic Indexing

Bei dieser Methode werden die Dokumente bzw. die Indexterme dieser in eine Matrizenform transformiert. Der Suchanfragevektor wird als Pseudo-Dokument ebenfalls in diese Matrix inkludiert. Jedes Element in der Matrix steht für einen gewissen Term eines Dokuments und der Wert dieses Elements ist sein vorher berechnetes Gewicht. Die Idee hinter diesem Konzept besteht nun darin, mit Hilfe von algebraischen Methoden die Hauptkomponenten einzelner Dokumente zu finden. Eine solche Methode ist die Singulärwertzerlegung. Diese ermöglicht es, die Singulärwerte einer Matrix zu berechnen, indem die Matrix in drei Matrizen aufgespalten wird. Das Produkt dieser drei Matrizen ergibt wiederum die Ursprungsmatrix (vgl. Landauer, Foltz & Laham, 1998; Baeza - Yates & Ribeiro - Neto , 1999).

Das Hauptziel dieses Algorithmus ist es, die hochdimensionalen Datenvektoren der einzelnen Dokumente, welche gleich den Spalten der Matrix sind auf, eine kleinere Dimension überzuführen. Um das zu erreichen, werden nur die k

größten Singulärwerte verwendet und die anderen verworfen. Die danach resultierende Matrix, welche mit Hilfe der reduzierten Singulärwertmatrix und den entsprechenden Vektoren der beiden anderen Matrizen zurücktransformiert wird, ist eine Matrix mit Rang k, welcher wesentlich kleiner ist als der Rang der Ursprungsmatrix. Diese Matrix hat die Eigenschaft, dass diese die Ursprungsmatrix am besten approximiert. Durch bilden eines Abstandsmaßes, wie beispielsweise den Kosinus des Winkels zwischen den nunmehr reduzierten Dokumentvektoren und dem Suchanfragevektor, kann nun die Ähnlichkeit der Dokumente mit der Suchanfrage bestimmen werden (vgl. Landauer et al., 1998; Baeza - Yates & Ribeiro - Neto , 1999).

Der größte Vorteil dieser Methode ist es, dass auch jene Dokumente gefunden werden, die nicht die gleichen Worte beinhalten wie die Suchanfrage, sondern auch jene Dokumente, die sowohl vom Inhalt als auch der Bedeutung nach der Suchanfrage ähnlich sind. Dies ist dadurch erreichbar, dass bei diesem Algorithmus sowohl die Dokumente als auch die Suchanfrage auf ihre Konzepte reduziert werden. Dieser Effekt ermöglicht es, mithilfe dieser extrahierten Konzepte die Ähnlichkeit zu bestimmen und somit verbesserte Ergebnisse zu erhalten. Diese Methode weist allerdings den großen Nachteil auf, dass es nicht möglich ist, die eventuell unterschiedlichen Bedeutungen von Wörtern korrekt zu erkennen (vgl. Landauer et al., 1998; Baeza - Yates & Ribeiro - Neto , 1999).

3.4.7 Neuronale Netzwerke

Neuronale Netze sind mehrdimensionale Graphen, die das menschliche Gehirn bzw. die Vorgänge im Gehirn auf einem Computer abbilden sollen. Die Knoten in diesem Graph repräsentieren die Neuronen, die Kanten, welche die Knoten verbinden, repräsentieren die synaptischen Verbindungen der Neuronen. Diese Kanten haben ein bestimmtes Gewicht und verändern das vom Sendeneuron ausgehende Signal für das Empfängerneuron ihrem Gewicht entsprechend. Der Zustand eines Knotens, der sogenannte *Activation Level*, ist abhängig vom Anfangszustand und vom eingehenden Signal und wirkt sich auch direkt auf das vom Knoten anschließend emittierte Signal aus (Baeza - Yates & Ribeiro - Neto, 1999).

Beim Information Retrieval besteht ein solches neuronales Netzwerk aus Dokumentknoten, aus Dokument – Termknoten und aus den Suchanfrage – Termknoten (vgl. Abbildung 2).

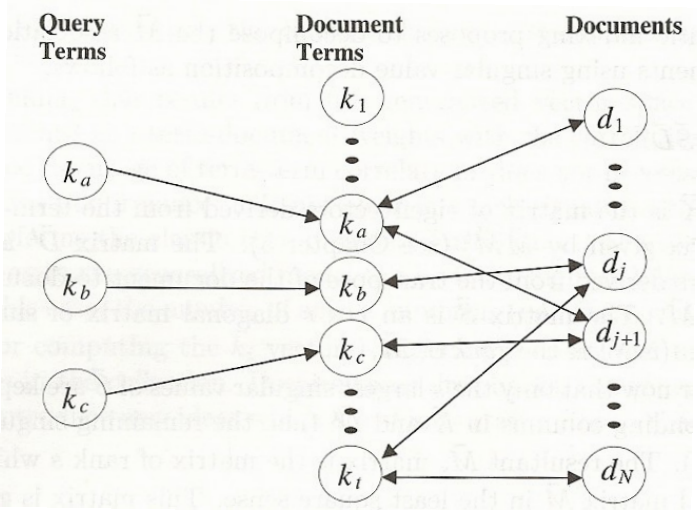

Abbildung 2: Beispiel für ein neuronales Netz (Baeza - Yates & Ribeiro - Neto , 1999).

Die Suchanfrage - Termknoten bekommen als Activation Level den Wert eins zugeordnet und senden ein Signal an die Dokument – Termknoten. Diese wiederum können je nach Zustand und Inputsignal, welches gleich dem Ausgangssignal der Suchanfrage – Termknoten ist, ein neues Signal generieren und an die Dokumentknoten weitersenden. Die Signale werden natürlich von den Gewichten der Kanten beeinflusst. Alle Signale, welche an einem Dokument ankommen, werden aufsummiert und die Summe dieses Gewichtes spiegelt die Relevanz diese Dokuments für die Suchanfrage wider. Das Ergebnis ist zu diesem Zeitpunkt dasselbe wie beim klassischen Vektorraummodell (vgl. Abschnitt 3.4.1) (vgl. Baeza - Yates & Ribeiro - Neto , 1999).

Um nun die Ergebnisse zu verbessern, wird mittels eines, von Benutzern gegebenes, Feedback eruiert, welche der Dokumente tatsächlich relevant sind. Anhand dieses Feedbacks werden die Gewichte der Verbindungen der Knoten adaptiert. Dieser Vorgang wird auch trainieren genannt und kann auch automatisch geschehen, indem das Netzwerk mit Hilfe einer Trainingsmenge, die sowohl relevante als auch nicht relevante Dokumente enthält, „lernt“ welche Eigenschaften für relevante Dokumente notwendig erscheinen und welche Eigenschaften nicht relevanten Dokumenten zuzuordnen sind. Wenn nun die Suchanfrage – Termknoten wiederum ein Signal aussenden, dann ändern sich die Signale die im Endeffekt an den Dokumenten ankommen aufgrund der geänderten Gewichte und ergeben dann aufsummiert eventuell ein andere Reihung der relevanten Dokumente (vgl. Baeza - Yates & Ribeiro - Neto; 1999, Grossman & Frieder, 2004).

3.4.8 Genetische Algorithmen

Genetische Algorithmen sind Algorithmen die auf Vererbung und Evolution beruhen. Die einzelnen Schritte eines solchen Algorithmus sind folgende (vgl. Grossman & Frieder, 2004; Lin, 1998; Lin, 1995; Lippe, 2005):

1. *Initialisierung der Population*: Dabei werden die ersten möglichen Lösungen für ein Problem ermittelt und dazu eine sogenannte Fitnessfunktion erstellt, die es ermöglichen soll zu beurteilen, wie gut die einzelnen Lösungen sind.

2. *Evaluation*: Dabei wird die relative Fitness der einzelnen Lösungskandidaten ermittelt. Fitness bedeutet dabei, wie gut die Lösung das vorgegebene Ziel erreicht.

3. *Selektion*: Eine bestimmte Anzahl von Lösungen wird ausgewählt, wobei Lösungen mit einer guten Fitness mit größerer Wahrscheinlichkeit ausgewählt werden.

4. *Reproduktion*: Mit Hilfe von *Rekombination*, also das vermischen der Eigenschaften von mehreren Ausgewählten Lösungen und von *Mutation* (zufällige Veränderung der Merkmale der Lösungen) wird nun eine neue Population erzeugt.

5. *Konvergenz*: Die Schritte Evaluation, Selektion und Reproduktion werden solange wiederholt, bis die Fitness der Lösungen gegen ein vorher bestimmtes Abbruchskriterium konvergiert.

Im Information Retrieval können genetische Algorithmen unter anderem dazu benutzt werden, geeignete Repräsentation von Dokumenten zu finden. Dazu erstellte man viele Repräsentationen, die sich allesamt auf von Benutzern ausgewählten Wörtern beziehen. Dazu wird dann eine Menge von Suchanfragen erstellt, die als Fitnessmaß dienen. Mit Hilfe der oben beschriebenen Schritte wird nun die Dokumentrepräsentation so lange verändert, bis sie eine entsprechende Fitness in Bezug auf die Suchanfragen aufweisen. Als Ergebnis erhält man dann geeignete Repräsentationen für ein Dokument (vgl. Grossman & Frieder, 2004).

Darüber hinaus sind diese Dokumente auch für das traditionelle Ziel des Information Retrieval geeignet. Dabei wird eine passende Repräsentation für die Dokumente einer Sammlung erzeugt und anschließend verändert ein generischer Algorithmus solange die Menge der Dokumente, bis die Fitness der ausgewählten Dokumente zufriedenstellend ist. Dass heißt die Iterationen finden solange statt, bis für die Suchanfrage genügend geeignete Dokumente gefunden werden konnten. (vgl. Grossman & Frieder, 2004).

3.4.9 Fuzzy Set Retrieval

Ein Fuzzy Set ist nach Zimmermann (2001) Menge, deren Element aus einer großen Menge von Elementen ausgewählt werden und deren Zugehörigkeit zu einer Menge durch eine sogenannte Zugehörigkeitsfunktion beschrieben wird. Bei einem normalisierten Fuzzy Set können die Werte dieser Zugehörigkeitsfunktion alle Werte im Intervall [0,1] annehmen. Wenn einem Element durch diese Zugehörigkeitsfunktion der Wert 0 zugewiesen wird, dann gehört das Element nicht zur Klasse, wohingegen Elemente, die den Wert 1 zugwiesen bekommen vollständig zur Klasse gehören. (vgl. Zimmermann, 2001; Baeza - Yates & Ribeiro - Neto , 1999).

Im Information Retrieval werden Fuzzy Sets oftmals in Verbindung mit Booleschen Modell (vgl. 3.4.4) verwendet. Dabei werden aus den zu untersuchenden Dokumenten Fuzzy Sets gebildet. Dies geschieht mit dazu geeigneten Methoden, welche die Zugehörigkeit der einzelnen Wörter zum Dokument ermitteln. Eine Methode ist beispielsweise jene, die Häufigkeiten der Wörter eines Dokumentes zu eruieren und diese dann mittels der Anzahl der Wörter im Dokument zu normieren. Anschließend wird für diese Fuzzy Sets die Relevanz zur Suchanfrage gebildet. Dafür werden beispielsweise algebraischen oder booleschen Methoden eingesetzt (vgl. Baeza - Yates & Ribeiro - Neto, 1999; Grossman & Frieder, 2004).

Laut den Autoren Baeza - Yates und Ribeiro - Neto (1999) sind Information Retrieval Methoden, welche auf das Fuzzy Set Modell aufbauen wenig beliebt und werden daher auch selten verwendet.

3.5 Informationsextraktion

Die Informationsextraktion ist ein Teilgebiet der maschinellen Sprachverarbeitung, das mit dem Thema der Arbeit eng in Verbindung steht. Die Hauptaufgabe der Informationsextraktion ist es, Texte zu analysieren um relevante Inhalte zu extrahieren. Die Autoren Moens und De Busser (2006) definieren in sehr abstrakter Weise Informationsextraktion wie folgt:

„*Informationsextraktion ist die Identifikation von speziellen Informationen aus unstrukturierten Daten, wie natürlich sprachlichen Text, sowie die daraus folgende simultane Klassifikation und Strukturierung dieser Daten in semantische Klassen, um diese Information der Informationsverarbeitung zugänglich zu machen*."[1]

[1] *Information extraction is the identification, and consequent or concurrent classification and structuring into semantic classes, of specific information found in unstructured data sources, such as natural language text, making the information more suitable for information processing tasks." (Moens & De Busser, 2006, Seite 2)*

Nach Neumann (2001) ist das Ziel der Informationsextraktion das extrahieren und strukturieren von domänenspezifischen Informationen aus Texten. Dabei sollen nicht die gesamten Dokumente, sondern gezielt jene Textpassagen, welche die relevanten Informationen beinhalten, untersucht werden. Um die Relevanz bestimmen zu können, bedarf es jedoch vordefinierter Regeln, die dem System zur Verfügung gestellt werden müssen (vgl. Neumann, 2001).

Moens und De Busser (2006) entgegnen diesen Ansatz der domänenspezifischen Informationsextraktion jedoch, dass ein ideales System domänenunabhängig sein soll oder zumindest die Überführung in eine andere Domäne ohne großen Aufwand durchführbar sein soll.

Die klassische Struktur eines Informationsextraktionsmodells zeigen die Autoren Turmo, Ageno und Català (2006) (vgl. Abbildung 3).

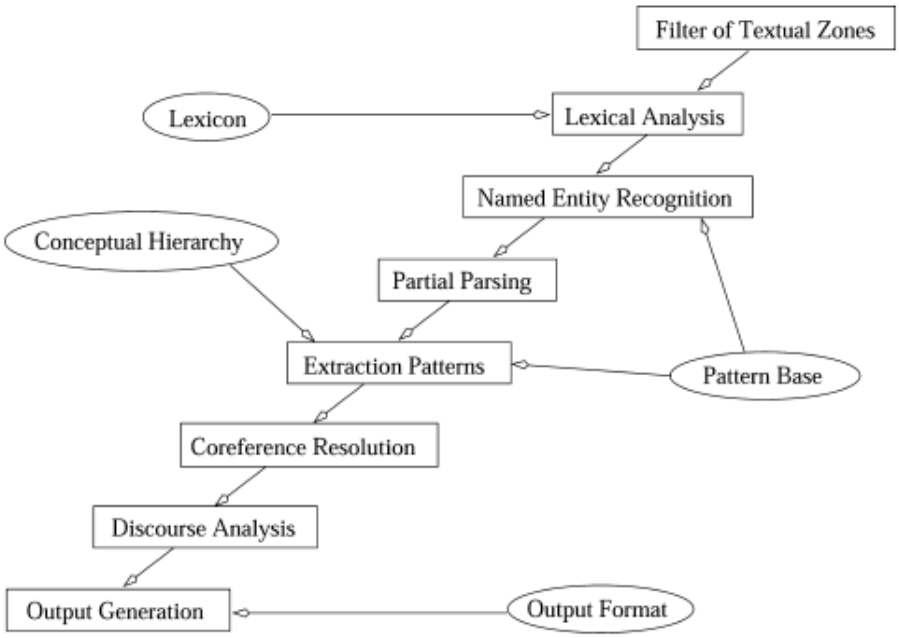

Abbildung 3: Informationsextraktion Architektur (Turmo, Ageno & Català 2006)

Viele Ansätze der Informationsextraktion folgen der in Abbildung 3 dargestellten Architektur obgleich diese natürlich in vielen Systemen in modifizierter Art und Weise Anwendung finden. Im Zuge ihrer Analysen extrahierten Turmo et al. (2006) allerdings Verarbeitungsschritte, die nahezu in allen Systemen zur Informationsextraktion verwendet werden. Diese Schritte sind Folgende:

1. *Textvorverarbeitung*: Die Textvorverarbeitung wurde bereits in Abschnitt 2.2.1 näher erläutert und findet auch im Bereich der Information Extraktion in ähnlicher Form Berücksichtigung.

2. *Syntaktische Parsing und semantische Interpretation*: In diesem Schritt werden Konzepte extrahiert, die den Inhalt eines Dokumentes widerspiegeln sollen. Dabei wird nicht vollständiges Parsing verwendet, sondern partielles Parsing. Mit einem partiellen Parsing wird versucht, die Nachteile, die ein vollständiges Parsing mit sich bringt (z.B.: geringe Robustheit, großer Aufwand, Mehrdeutigkeiten treten auf usw.), zu umgehen, indem nur sich nichtüberlappende Phrasen berücksichtigt werden. Dieser Vorgang wird oft auch Chunking genannt (vgl. 2.2.1.4). Nach diesem Schritt werden nun die Abhängigkeiten der Phrasen bezüglich der verschiedenen Domänen aufgelöst, um so eine semantische Interpretation zu ermöglichen. Für diesen Vorgang gibt es 2 verschiedene Ansätze. Einerseits das sogenannte Pattern Matching, welches mit Hilfe von vorgefertigten domänenspezifischen Schablonen versucht, solche Abhängigkeiten aufzulösen. Anderseits gibt es den grammatikalischen Ansatz, wobei unter Zuhilfenahme von vordefinierten grammatikalischen Relationen und Regeln versucht wird, eine domänenabhängige semantische Deutung zu.

3. *Diskurs Analyse*: Diese Analyse dient dazu, Abhängigkeiten zu finden, die sich über Satzgrenzen hinaus und oftmals auch über mehrere Absätze erstrecken (vgl. Abschnitt 2.1.3). Ein wesentlicher Bestandteil dieser Diskursanalyse ist die in Abschnitt 2.2.1.5 vorgestellte Koreferenzauflösung.

4. *Erzeugung des Ausgangsformats*: Hierbei werden die extrahierten Informationen in das gewünschte Ausgangsformat übergeführt, um damit eine weitere Verarbeitung dieser Daten zu.

Die in diesem Abschnitt vorgestellten Verfahren sind klassische Aspekte der Informationsextraktion. In den letzten Jahren hat maschinelles Lernen im Bereich der Informationsextraktion immer mehr an Bedeutung gewonnen. Dabei wird anhand von vor erzeugten Trainingsmengen versucht, automatisch Templates zu generieren, die dann für die Informationsextraktion eingesetzt werden können (vgl. Neumann, 2001; Turmo et al., 2006).

3.6 Konzeptextraktion

Konzeptextraktion ist nach Moens und Angheluta (2003) die Identifizierung konzeptrelevanter Terme und Phrasen aus Texten, sowie eine mögliche Überführung dieser in allgemeinere, abstrakte Konzepte (vgl. Moens & Angheluta, 2003).

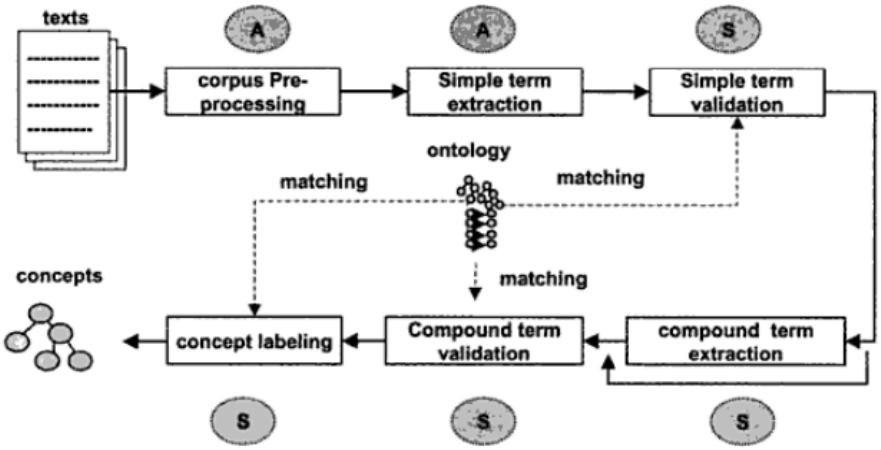

Abbildung 4: Schema der Konzeptextraktion (vgl. Roussey et.al, 2006)

Die Autoren Roussey, Calabretto, Harrathi & Gammoudi (2006) zeigen eine schematische Darstellung einer Konzeptextraktion aus multilingualen Korpora (vgl. Abbildung 4). Dabei ist zu beachten, dass hierbei auch einige der Schritte als semi – automatisch angeführt sind, die allerdings prinzipiell auch vollautomatisch ausgeführt werden könnten.

Im Preprocessing Schritt wird sowohl eine Tokenisierung durchgeführt als auch die Worthäufigkeit ermittelt. Dabei wird eine Liste erstellt, die alle Wörter, die dazugehörige Häufigkeit, sowie die Position der Wörter enthält. Im nächsten Schritt werden einzelne Wörter extrahiert und anschließend erfolgt eine Validierung um eine Auswahl der potentiellen Kandidaten zu treffen. Danach werden Terme, die aus mehreren Wörtern bestehen (*compound term*), gebildet. Diese Phrasen werden aus den in den vorangegangen Schritten extrahierten Wörtern gebildet und ebenfalls einer Validierung unterzogen. Dadurch werden die nicht geeigneten Phrasen entfernt. Im Abschließenden *concept labeling* werden die einzelnen Phrasen und Wörter in Abhängigkeit ihrer semantischen Ähnlichkeiten gruppiert, um die einzelnen Konzepte zu erhalten. Für diese Konzepte wird danach eine Bezeichnung ermittelt und dem Konzept zugeordnet (vgl. Roussey et.al, 2006).

3.7 Ontologieextraktion

Der Autor Gruber (1993) definiert Ontologie als "*explizite Spezifikation einer Konzeptualisierung*". Eine Ontologie ist also ein abstraktes Modell eines Wissensbereiches (knowledge domain). Dabei werden die Eigenschaften von wichtigen Konzepten und Begriffen und deren Beziehungen zueinander modelliert. Diese Formalisierung soll in einer eindeutigen Sprache erfolgen, welche aus einem standardisierten Vokabular besteht und sowohl von Menschen als auch von

Maschinen verstanden werden kann (vgl. Baader, Horrocks, & Sattler, 2004; Hesse, 2002).

Die Ontologieextraktion befasst sich mit der Extraktion von Konzepten und Modellen aus Texten, Dokumenten und Internetseiten. Abbildung 5 zeigt eine schematische Darstellung der Extraktion von Ontologien aus unstrukturierten Texten.

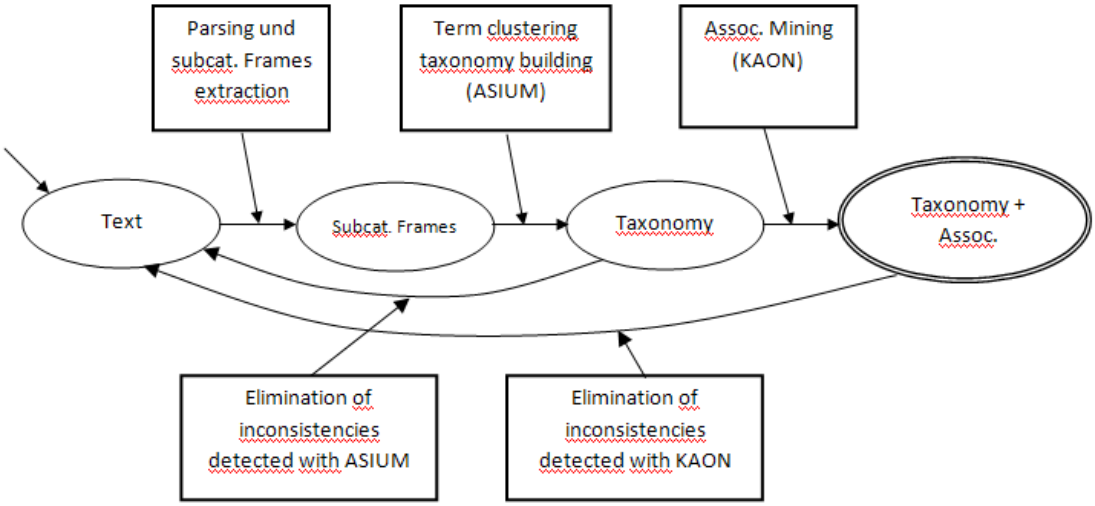

Abbildung 5: Ontologieextraktion (Kof & Pizka, 2005, überarbeitet)

Nach den Autoren Kof und Pizka (2005) erfolgt die Extraktion dabei in vier wesentlichen Schritten. Zu Beginn wird eine Termextraktion durchgeführt, wobei jeder Satz einzeln analysiert wird und anschließend Hauptwortphrasen als mögliche Konzepte identifiziert werden. Im zweiten Schritt werden aus den einzelnen Phrasen Cluster gebildet. Dabei wird eine Phrase einem Cluster zugeordnet, wenn diese Phrase eine Verbindung zum Cluster aufweist. Diese Cluster werden dann im nächsten Schritt zu größeren Clustern vereinigt, falls diese gemeinsam Konzepte (Phrasen) beinhalten, um so allgemeinere Beschreibungen dieser Konzepte, welche durch die Cluster repräsentiert werden, zu ermöglichen. Abschließend werden noch die Verbindungen zwischen den einzelnen Cluster ermittelt und danach Ontologien aus dem Dokument erzeugt (Kof & Pizka, 2005).

Abbildung 6 zeigt ein Beispiel für eine Ontologie. Dabei repräsentieren die Kreise die Begriffe und die Pfeile die Verbindungen der einzelnen Begriffe. Für die formelle Beschreibung solcher Ontologien gibt es spezielle Sprachen. Nach Lehner (2008) sind diese das Ressource Description Framework (RDF), die Web Ontologie Language (OWL), DAML+OIL, XTM und F-Logic (vgl. Lehner, 2008).

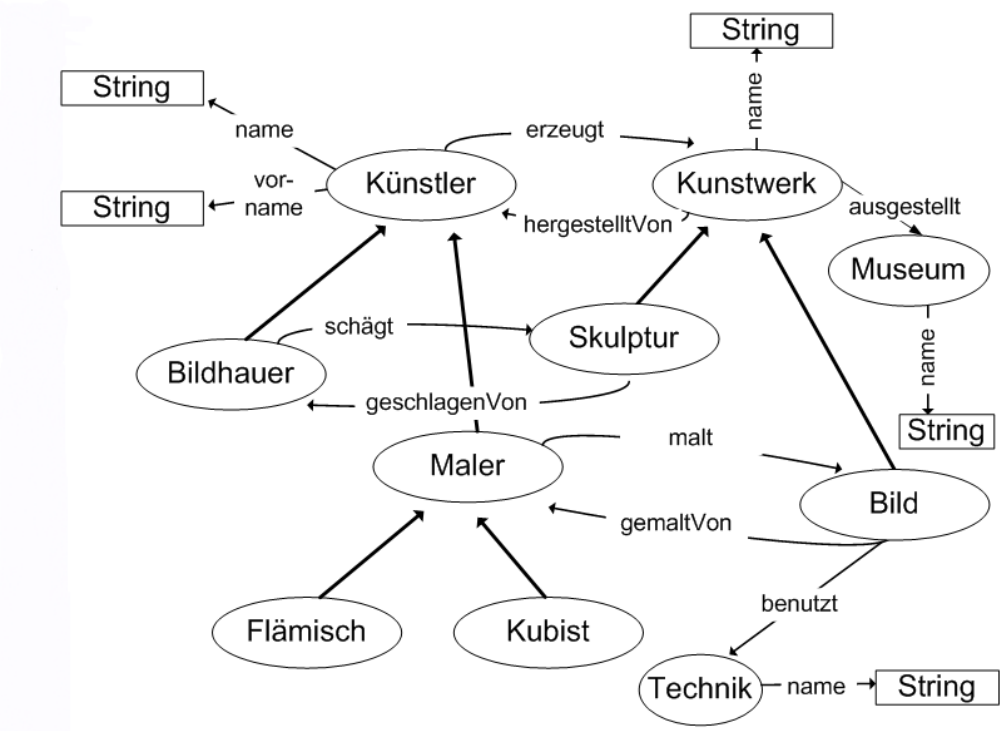

Abbildung 6: Beispiel für eine Ontologie (Wikipedia, 2010)

3.8 Topic Segmentation

Topic Segmentation ist das Auffinden und die Strukturierung von Texten anhand von jenen Themengebieten, die der Text repräsentiert, sowie das Auffinden von Textsegmenten, die zu einem bestimmten Thema gehören (vgl. Moens, Angheluta & De Busser, 2003). Das bedeutet, dass Topic Segmentation die Aufgabe hat, Änderungen von Themen innerhalb eines Textes aufzuspüren, die Grenzen dieser einzelnen Themen zu definieren und Verbindungen zwischen diesen Segmenten zu identifizieren (vgl. Ferret, 2002). Das Hauptproblem beim Topic Segmentation ist es, eine Definition zu finden, wann sich ein Thema ändert bzw. Methoden zu finden, die die Zugehörigkeit zu einem Thema definieren (vgl. Moens et al., 2003; Beeferman, Berger & Lafferty, 1999).

Der bekannteste Topic Segmentation Algorithmus ist nach Moens et al. (2003) der TextTiling Algorithmus von Hearst (1997). Dieser Algorithmus arbeitet in drei Arbeitsschritten. Diese Schritte sind Tokenisierung, lexikalische Analyse und Identifizierung der Themengrenzen und werden nachfolgend kurz erläutert:

1. *Tokenisierung*: Zusätzlich zu der in Abschnitt 2.2.1.1 vorgestellten Tokenisierung werden beim TextTiling Algorithmus noch weitere Schritte durchgeführt, die allesamt in den Bereich der Textvorverarbeitung (vgl. Abschnitt 2.2.1) fallen. Dabei werden nur die tatsächlichen Textteile

verwendet, wohingegen Wörter in Überschriften, Titel etc. verworfen werden. Danach werden Stoppwörter entfernt und die übrigen Wörter werden mittels eines Stemming Algorithmus auf ihre Grundform reduziert. Danach wird, unabhängig von der tatsächlichen Satzstruktur, der Text in Pseudo – Sätze einer bestimmten Länge unterteilt (vgl. Hearst & Plaunt, 1993; Hearst, 1997).

2. *Lexikalische Analyse*: In diesem Schritt werden nun Blöcke, die aus einer bestimmten Anzahl von Pseudo – Sätzen bestehen auf ihre lexikalische Ähnlichkeit überprüft. Dafür gibt es zwei Methoden: Einerseits gibt es die Blockmethode, bei der immer die Ähnlichkeit zwischen zwei Blöcken berechnet wird, indem das normalisierte innere Produkt der Vektoren, die diese Blöcke repräsentieren, berechnet wird. Je größer dieser Wert, desto ähnlicher sind sich diese Blöcke. Die einzelnen Elemente dieser Vektoren sind die Wörter bzw. die Gewichte dieser Wörter, wobei die Gewichte die Häufigkeit des Wortes in einem Block repräsentieren. Andererseits gibt es die sogenannte *Vocabulary Introduction*, deren Grundprinzip darin besteht, die Anzahl von neuen Wörtern, die in einem neuen Block auftreten, zu ermitteln und anschließend mit der Länge des Segmentes zu normieren. Ein Auftreten von vielen neuen Wörtern ist ein Indiz für einen Themenwechsel (vgl. Hearst & Plaunt, 1993; Hearst, 1997).

3. *Identifizierung der Themengrenzen:* Die Bestimmung der Grenzen der einzelnen Themen eines Textes funktioniert anhand von der bei der lexikalischen Analyse ermittelten Ergebnisse. In Abbildung 7 wird ein beispielhaftes Ergebnis einer derartigen Analyse dargestellt. Auf der X-Achse sind die einzelnen Sätze aufgetragen, auf der Y-Achse die Ähnlichkeiten zwischen diesen Pseudo - Sätzen. Um die einzelnen Themengrenzen identifizieren zu könne, müssen Sätze gefunden werden, die wenig gemeinsam haben (Täler der Kurve in Abbildung 7). Wenn nun die Kurve vor solch einem Tal sehr stark abfällt und danach wieder sehr steil ansteigt, dann ist die Wahrscheinlichkeit sehr groß, dass in diesem Punkt ein Abschnittswechsel stattfindet. Dies impliziert, dass sich zwischen diesen Sätzen die Themengrenze befindet (vgl. Hearst & Plaunt, 1993; Hearst, 1997).

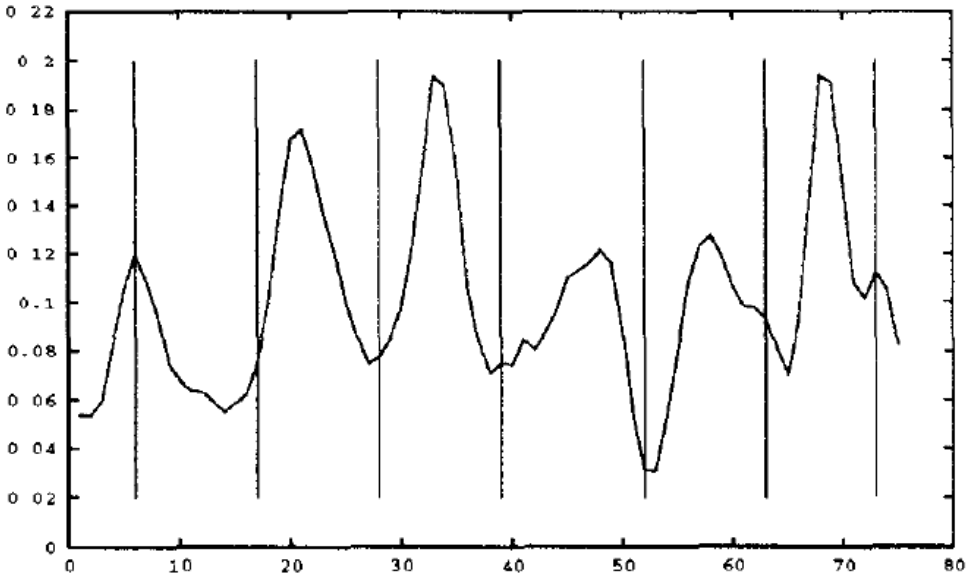

Abbildung 7: Auswertung der Ergebnisse von TextTiling (Hearst & Plaunt, 1993)

Topic Segmentation wird in vielen Bereichen des Natural Language Processings, wie beispielsweise automatische Textzusammenfassung, Information Retrival usw., eingesetzt (vgl. Hearst & Plaunt, 1993; Hearst, 1997).

Neben dem TextTiling Algorithmus gibt es noch viele weitere Ansätze für das Topic Segmentation. Einen Überblick über diese Algorithmen liefern beispielsweise Sun, Mitra, Zha, Giles & Yen (2007), Choi (2000) und Ferret (2002).

3.9 Zusammenfassung

In diesem Kapitel wurden einige wichtige Teilgebiete der maschinellen Verarbeitung natürlicher Sprache vorgestellt und näher erläutert. Dabei wurde hauptsächlich auf jene Teilgebiete eingegangen, die auch mit dem Thema der Arbeit in Verbindung stehen.

Das der Extraktion von semantisch relevanten Daten aus natürlich sprachlichen Texten in Hinblick auf eine automatische Fragengenerierung am ähnlichsten Teilgebiet des NLP ist die automatische Textzusammenfassung. Dabei ist es ebenfalls das Ziel, Daten zu extrahieren, die den gesamten Text repräsentieren. Daher können in weiterer Folge viele Ansätze aus diesem Teilgebiet übernommen werden. Aus den Gebieten der Konzept- bzw. der Ontologieextraktion können vor allem die semantischen Analysemechanismen übernommen werden, um semantische Ähnlichkeiten zwischen Wörtern, Phrasen und Sätzen zu ermitteln. Das Information Retrieval liefert ebenfalls viele Erkenntnisse über das Auffinden bzw. das Berechnen von Ähnlichkeiten zwischen Wörtern und Dokumenten. Darüber hinaus werden im Information Retrieval vielfältige Methoden aufgezeigt, wie

man die Ergebnisse sortieren bzw. reihen kann, und es werden auch einige Verfahren zur Termgewichtung dargeboten. Topic Segmentation kann dazu verwendet werden Strukturen in unstrukturierte Texte einzufügen, um so die zentralen Themen eines Textes besser herausfiltern zu können. Die Textklassifizierung hingegen kann benutzt werden, um die Extraktion domänenabhängig zu gestalten. Diese würde mit großer Wahrscheinlichkeit die einzelnen Ergebnisse verbessern, allerdings würde das Einsatzgebiet des Systems doch beträchtlich eingeschränkt.

Um weitere Erkenntnisse im Bereich des Natural Language Processing, in Bezug auf das Thema dieser Arbeit, zu gewinnen, wird im nächsten Kapitel näher auf den aktuellen Forschungsstand in diesem Bereich eingegangen.

4 Aktueller Forschungsstand

In diesem Kapitel wird der aktuelle Forschungstand einiger Teilgebiete der maschinellen Verarbeitung natürlicher Sprache, welche in Verbindung zum Thema dieser Arbeit stehen, aufgezeigt. Hierbei wird versucht, wichtige Erkenntnisse zu gewinnen, welche für eine Extraktion von semantisch relevanten Daten in Hinblick auf eine automatische Fragengenerierung von Bedeutung sind. Die ermittelten Erkenntnisse sollen mit dem Ziel dienen, die Extraktion von Wörter, Phrasen und Sätze, welche die für eine automatische Generierung von Fragen benötigt werden, zu ermöglichen.

4.1 Extraktion inhaltsrelevanter Daten

In diesem Abschnitt werden Forschungsansätze aufgezeigt, die eine automatische Extraktion von Wörtern (engl. keywords), Phrasen (engl. keyphrases) und Sätzen zum Inhalt haben. Diese extrahierten Daten sollen dabei eine Repräsentation des Inhaltes der Texte ermöglichen.

4.1.1 KEA - Keyword Extraction Algorithmen

Der von Witten, Paynter, Frank, Gutwin und Nevill-Manning (1999) entworfene Algorithmus extrahiert Schlüsselwörter bzw. – Phrasen mit Hilfe von maschinellem Lernen, insbesondere des Naiven Bayes Klassifikators (vgl. Abschnitt 3.2). Dabei wird mittels einer Trainingsmenge ein Modell für die Identifizierung dieser Phrasen gebildet. Die Trainingsmenge besteht dabei aus einer Menge von Dokumenten, deren Schlüsselwörter und Schlüsselphrasen vom Autor selbst angegeben werden. Das Trainieren des Models erfolgt dabei anhand von zwei Merkmalen, einerseits die *TFxIDF* und andererseits die *first occurence*. Die TFxIDF repräsentiert dabei die relative eines Wortes im Dokument. Dabei wird die Häufigkeit aller Wörter im Dokument ermittelt und mit der Häufigkeit der Wörter in einem repräsentativen Korpus normiert. Dadurch ist es möglich, Wörter, welche in vielen Dokumenten vorkommen und daher im Normalfall eine geringe Aussagekraft aufweisen, zu erkennen und gegebenenfalls nicht zu berücksichtigen. Die Berechnung erfolgt nach Formel 4.1, wobei $freq(P,D)$ die Anzahl der Vorkommen der Phrase P im Dokument D, $Size(D)$ die Anzahl aller Wörter in D, $df(P)$ die Anzahl der Dokumente im Korpus, welche die Phrase P beinhalten und N die Anzahl der Dokumente im Korpus darstellen (vgl. Witten et al., 1999).

$$TFxIDF = \frac{freq(P,D)}{Size(D)} * -\log_2 \frac{df(P)}{N} \qquad 4.1$$

Das zweite Merkmal, die *first occurence,* beschreibt die Position des ersten Auftretens der Phrase im Dokument und wird berechnet, indem der Quotient der Anzahl der Wörter, die vor der Phrase auftreten, mit der Anzahl aller Wörter im Dokument gebildet wird (vgl. Witten et al., 1999).

Um diese Merkmale extrahieren zu können sind nun einige Vorverarbeitungsschritte notwendig. Zu Beginn findet ein *input cleaning* statt wobei der Text tokenisiert wird. Dabei werden Satzzeichen, Klammern und Zahlen durch Phrasengrenzen ersetzt, Anführungszeichen gelöscht und mit Bindestrich versehene Wörter aufgetrennt. Darüber hinaus werden alle weiteren Zeichen, die keine Buchstaben sind entfernt. Anschließend findet die Identifizierung der Phrasen statt. Dabei gibt es allerdings die Einschränkungen, dass diese Phrasen eine bestimmte Länge nicht überschreiten, nicht mit Stoppwörtern beginnen und keine Eigennamen sein dürfen (z.B.: keine einzelnen Wörter, die mit Großbuchstaben beginnen). Abschließend wird noch ein Stemming Algorithmus verwendet, um die unterschiedlichen Flexionsformen der Wörter zu berücksichtigen (vgl. Witten et al, 1999).

Mit Hilfe der Formeln 4.2 und 4.3 wird danach die Wahrscheinlichkeit, dass eine Phrase eine Schlüsselphrase ist, berechnet.

$$P(yes) = \frac{Y}{Y+N} P_{TFxIDF}(t|yes)\, P_{distance}(d|yes) \tag{4.2}$$

$$p = \frac{P(yes)}{P(yes) + P(no)} \tag{4.3}$$

Dabei entspricht dabei Y der Anzahl der positiven Vorkommen in den Trainingsdokumenten, N jener der negativen Vorkommen. $P_{TFxIDF}(t|yes)$ ist die Anzahl der Phrasen in den Trainingsdokumenten, welche Schlüsselwörter sind und gleichzeitig den Wert t (*TFxIDF*) annehmen. $P_{distance}(d|yes)$ hingegen, ist die Anzahl jener Phrasen in den Trainingsdokumenten, die Schlüsselwörter sind und für die die Zahl d als *first occurence* ermittelt wird. $P(no)$ ist der gegenteilige Ausdruck zu $P(yes)$. Der Wert p gibt dann die Wahrscheinlichkeit, dass diese Phrase eine Schlüsselphrase ist, an. Anhand dieser Wahrscheinlichkeit werden die Phrasen gereiht. Anschließend werden jene Phrasen eliminiert, die Teilphrasen einer höher gereihten Phrase sind. Abschließend werden dann die n besten Phrasen als Schlüsselphrasen ausgewählt, wobei n die Anzahl der gewünschten Phrasen darstellt (vgl. Witten et al., 1999).

Die Evaluierung der Ergebnisse des KEA zeigt sich, dass durchschnittlich ein bis zwei Schlüsselwörter (bei insgesamt 5 Auszuwählenden) mit denen von den Autoren ausgewählten Schlüsselwörtern ident sind. Dies ist nach Witten et al.

(1999) ein relativ guter Wert, vor allem dann, wenn man den Umstand berücksichtigt, dass zwei verschiedene Personen bei der Auswahl von Schlüsselwörtern eine ähnliche Übereinstimmung erreichen. (vgl. Witten et al., 1999).

Der Autor Turney (2003) erweiterte diesen Algorithmus dahingehend, dass die Kohärenz des Textes mit einbezogen wird. Dabei wird der statistische Zusammenhang unter den möglichen Schlüsselwörtern ermittelt, da Wörter, welche zueinander eine semantische enge Verbindung aufweisen, oftmals gemeinsam vorzukommen. Ziel dieser Methode ist es, die Schlüsselwortsuche domänenunabhängiger zu gestalten und qualitativ höherwertige Schlüsselörter zu extrahieren (vgl. Turney, 2003).

Turney (2003) verwendet bei seinen Erweiterungen zu KEA die Internet - Suchmaschine AltaVista um die Zusammenhänge der Wörter zu ermitteln, indem er die Anzahl der zurückgelieferten Suchanfragen als Indikator für diese Beziehungen benutzt. Dabei werden speziell formulierte Suchanfragen, die die Ergebnisse des Suchalgorithmus einschränken, verwendet. Turney (2003) ermittelt dafür zwei Werte. Als erstes ermittelt er die Anzahl der Treffer zweier bestimmter Wörter, die gemeinsam miteinander auf einer Webseite vorkommen, mit der Einschränkung, dass ein Wort mindestens zehn Mal vorkommen muss. Der zweite Wert, der mit AltaVista ermittelt wird, ist die Anzahl der Seiten, bei denen ein Wort in einer Überschrift oder im Titel und das andere Wort im *body* der Seite vorkommen. Das wird dadurch bewerkstelligt, indem das Wort mit großen Anfangsbuchstaben in die Suchanfrage integriert wird, da großgeschriebene Wörter ein Indiz für das Vorkommen in Titel oder Überschrift sind. Diese beiden ermittelten Werte werden dann in das Auswahlverfahren geeignet integriert (vgl. Turney, 2003).

Einen Vergleich der Ergebnisse der einzelnen Feature Sets, die bei KEA anwendbar sind, zeigt Abbildung 8. Dabei ist zu beachten, dass die Basisvariante (*Baseline Feature Set*) des KEA Algorithmus jenes Set ist, welche *TFxIDF* und *first occurence,* wie zu Beginn dieses Abschnittes beschrieben, nutzt. Das *Coherence Feature Set* verwendet nur die Erweiterung von Turney (2003) zur Ermittlung der Schlüsselphrasen. Beim *Keyphrase Feature Set* wird zusätzlich zur Basisvariante noch die Schlüsselwörterfrequenz berücksichtig, welche die Häufigkeit repräsentiert, dass diese Phrase in einem Dokument des verwendeten Korpus als ein von einem Autor zugewiesenes Schlüsselwort auftritt. Das *Merged Feature Set* vereinigt sowohl die Basisvariante als auch die Coherence Feature Set und Keyphrase Feature Set. Wie in Abbildung 8 liefert das Merged Feature Set die besten Ergebnisse (vgl. Turney, 2003).

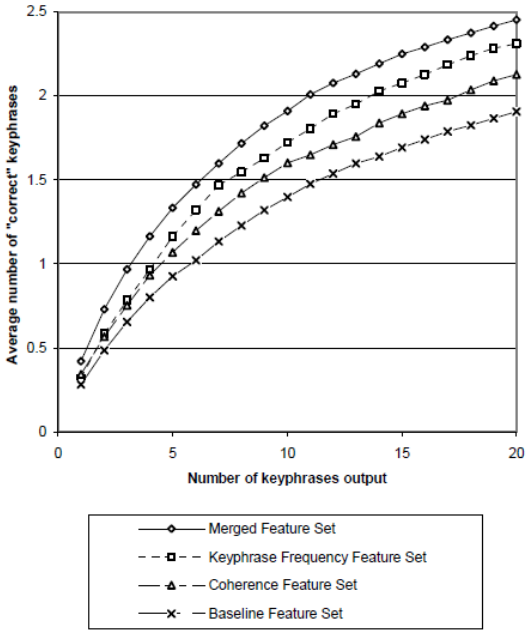

Abbildung 8: Vergleich der KEA - Variationen (vgl. Turney, 2003).

4.1.2 GenEx

Dieser von Turney (2000) entwickelte Algorithmus zur Extraktion von Schlüsselwörtern besteht im Wesentlichen aus zwei Teilen. Einerseits wird der *Genitor*, andererseits der *Extractor* verwendet. Der Genitor ist ein von Whitley (1989) entwickelter genetischer Algorithmus (vgl. Abschnitt 3.4.8), der in diesem System ausschließlich dazu verwendet wird, den Parametervektor, der zur Auswahl der Schlüsselphrasen herangezogen wird, durch ein Lernverfahren zu optimieren. Der Extractor ist ein Extraktionsalgorithmus der anhand von zwölf Parametern aus dem Eingangstext eine Liste von Schlüsselwörtern extrahiert (vgl. Turney, 2000).

Der Extraktionsalgorithmus arbeitet in zehn Schritten, die nachfolgend kurz erläutert werden. Die Bezeichnungen der zwölf Parameter sind dabei ausschließlich mit Großbuchstaben geschrieben (vgl. Turney, 2000):

1. *Erzeugung der Liste mit gestemmten Wörtern*: Als erstes werden alle Wörter, die weniger als drei Zeichen lang sind, eliminiert. Danach findet eine Grundformreduktion (vgl. 2.2.1.3) statt. Bei dieser Art der Grundformreduktion werden die Wörter allerdings einfach bei der Länge STEM_LENGTH abgeschnitten, da diese Form des Stemming wesentlich schneller und flexibler ist.

2. *Worthäufigkeit*: Die Anzahl der Vorkommen eines gestemmten Wortes im gesamten Text wird berechnet. Zusätzlich wird die Position des ersten Auftretens des Wortes ermittelt. Um nun ein Gewicht für ein Wort zu

erhalten, wird es, falls es vor dem Schwellwert FIRST_LOW_TRESH das erste Mal vorkommt mit dem Faktor FIRST_LOW_FACTOR multipliziert, falls diese Wort nach dem FIRST_HIGH_TRESH vorkommt wird es mit dem FIRST_HIGH_FACTOR multipliziert. Dadurch können Wörter, die an gewissen Positionen im Text das erste Mal auftauchen speziell gewichtet werden.

3. *Auswahl der besten Wörter*: In diesem Schritt werden die NUM_WORKING höchst gewichteten Wörter aus der Liste ausgewählt.

4. *Phrasenextraktion*: Aus dem Ursprungstext werden alle Sequenzen von bis zu drei aufeinanderfolgenden Wörtern als Phrase ausgewählt, unter der Bedingung, dass keine Stoppwörter zwischen den einzelnen Wörtern auftreten. Nach dem Auffinden werden auch die Wörter der Phasen, wie im ersten Schritt beschrieben, gestemmt.

5. *Ermittlung des Phrasengewichtes*: Die Phrasen werden analog zu den Wörtern gewichtet mit der Ausnahme, dass dieses ermittelte Gewicht in Abhängigkeit davon, ob die Phrase zwei bzw. drei gestemmte Wörter beinhaltet, mit dem Faktor FACTOR_TWO_ONE oder FACTOR_THREE_ONE multipliziert wird.

6. *Expansion*: Zu jedem Wort aus der Liste der NUM_WORKING Bestenliste wird jeweils die am höchsten gewichtete Phrase ermittelt, die dieses Wort beinhaltet und der Reihe nach in eine Phrasenliste eingefügt.

7. *Duplikate entfernen*: Falls eine Phrase zwei Mal in der Phrasenliste vorkommt, so wird die schlechter gereihte Phrase entfernt.

8. *Suffixe hinzufügen*: Für jede Phrase bzw. für jedes gestemmte Wort einer Phrase wird jenes Wort gesucht, das dieses gestemmte Wort beinhaltet und dazu am häufigsten im Text vorkommt, um anschließend das Suffix dieses Wortes dem gestemmten Wort anzufügen.

9. *Großbuchstaben hinzufügen*: In diesem Schritt wird versucht, die beste Schreibweise der Phrasen in Hinblick auf Groß/Kleinschreibung zu finden.

10. *Auswahl der geeigneten Phrasen*: Im abschließenden Schritt werden die NUM_PHRASES besten Phrasen ausgewählt. Dabei werden noch die Parameter SUPPRESS_PROPER und MIN_LENGTH_LOW_RANK berücksichtigt. SUPPRESS_PROPER kann 0 oder 1 annehmen und bestimmt somit, ob Eigennamen in den ausgewählten Phrasen vorkommen dürfen oder nicht. Alle endgültig ausgewählten Phrasen sollten länger sein als MIN_LENGTH_LOW_RANK, wobei die Länge der Phrasen aus der

tatsächlichen Länge der Phrase und der durchschnittlichen Länge aller Phrasen im Text berechnet wird. Eine Phrase, die kürzer ist kann aber dennoch ausgewählt werden, falls diese höher gereiht ist als MIN_RANK_LOW_LENGTH. Des Weiteren dürfen keine Verben und keine Stoppwörter in den ausgewählten Phrasen sein.

Für das Training bekommt der Genitor zwei Parameter vorgegeben, nämlich NUM_PHRASES zwischen fünf und fünfzehn und NUM_WORKING ist das Fünffache von NUM_PHRASES. Die restlichen zehn Parameter werden durch den Genitor selbst ermittelt (Turney, 2000).

Nach Turney (2000) werden rund 80% der durch GenEx extrahierten Phrasen von menschlichen Betrachtern als akzeptabel eingestuft.

4.1.3 Baseline Methoden

Die Autoren HaCohen-Kerner, Gross und Masa (2005) stellen in ihrem Paper *Automatic Extraction and Learning of Keyphrases from Scientific Articles* Methoden zur Extraktion von Schlüsselwörtern vor, welche auf grundlegende Merkmale der Texte beruhen. Sie versuchen dabei zu ermitteln, welches der einzelnen Merkmale sich am besten für das Auffinden von Schlüsselwörtern eignet. Des Weiteren wenden sie verschiedene Algorithmen des maschinellen Lernens an, um herauszufinden, welcher Algorithmus das beste Ergebnis erzielen kann (vgl. HaCohen-Kerner et al., 2005).

Die Autoren HaCohen-Kerner et al. (2005) verwenden dabei insgesamt elf auf verschiedenen Merkmalen basierende Methoden:

1. *Worthäufigkeit (TF)*: Die Anzahl der Vorkommen eines Wortes im gesamten Text. Die N häufigsten Wörter werden ausgewählt.

2. *Phrasenlänge (TL)*: Diese Länge entspricht der Anzahl der Wörter in den Phrasen. Die N längsten Phrasen werden dabei ausgewählt.

3. *Ersten N Phrasen (FN)*: Nur die ersten N Phrasen des Dokuments werden ausgewählt, da Autoren dazu tendieren, wichtige Informationen am Beginn einzufügen.

4. *Letzten N Phrasen (LN)*: Nur die letzten N Phrasen des Dokuments werden ausgewählt, aufbauend auf die Annahme, dass die Autoren die wichtigsten Schlüsselwörter in die Zusammenfassung integrieren und diese für gewöhnlich am Ende des Textes positioniert ist.

5. *Absatzbeginn (PB)* bzw*. Absatzende (PE)*: Die Phrasen und Wörter werden anhand der Position in einem Absatz gereiht. Dabei wird angenommen dass die wichtigsten Terme am Beginn bzw. am Ende eines Absatzes stehen.

6. *Resemblance to Title (RT)*: Die Phrasen werden anhand der Ähnlichkeit der Sätze, in denen sie vorkommen, mit dem Titel gereiht.

7. Maximal Section Headline Importance (MSHI): Die Phrasen werden anhand ihres wichtigsten Auftretens in Abschnitten mit bestimmten Überschriften, wie beispielsweise Einleitung, Kurzfassung oder Zusammenfassung, gereiht, da in diesen Abschnitten normalerweise die wichtigsten Informationen zusammengefasst sind.

8. *Accumulative Section Headline Importance (ASHI)*: Ähnlich zu MSHI nur werden hierbei alle Vorkommen in bestimmten Abschnitten berücksichtigt.

9. *Negative Brackets (NBR)*: Phrasen, die in Klammern angeführt sind, werden negativ bewertet.

10. *TF x MSHI*: Diese Methode vereinigt die Methode der Worthäufigkeit mit der MSHI Methode.

Durch ihre Untersuchungen fanden die Autoren HaCohen-Kerner et al. (2005) heraus, dass die Methode MSHI die besten Ergebnisse liefert. Weiters liefern TFxMSHI, TF, TN und ASHI ebenfalls zufriedenstellende Resultate. Die Analysen der maschinellen Lernalgorithmen zeigen, dass mit dem Algorithmus J48 die besten Ergebnisse erzielt werden können. Um nun die Ergebnisse weiter verbessern zu können, ist es erforderlich, eine möglichst gute Kombinationsmöglichkeit der einzelnen Methoden zu finden (vgl. HaCohen-Kerner et al, 2005).

4.1.4 Schlüsselwörterextraktion mittels Neuronaler Netze

Die Autoren Wang, Peng und Hu (2006) stellen ein System vor, das ein neuronales Netz benutzt, um Schlüsselwörter aus Texten zu extrahieren. Ein neuronales Netz (vgl. Abschnitt 3.4.7) besteht in diesem Fall aus einem Input-Layer, einem Hidden – Layer und einem Output – Layer (vgl. Abbildung 9).

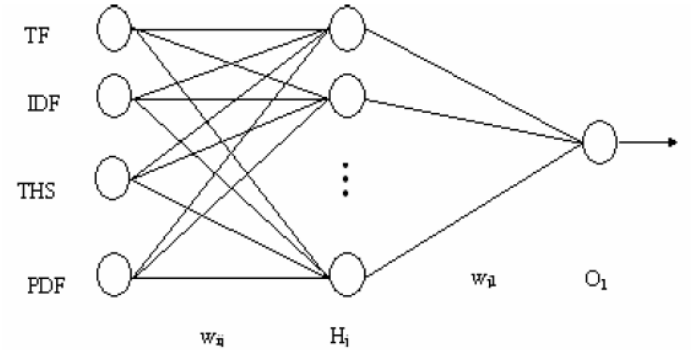

Abbildung 9: Neuronales Netzwerk für die Extraktion (vgl. Wang et al., 2006)

Dem Netzwerk werden vier verschiedene Merkmale einer Phrase aus einem Dokumente zugeführt. Diese vier Merkmale sind Termfrequenz tf, inverse Termfrequenz idf (vgl. Abschnitt 3.4.1) und jeweils die Anzahl der Absätze bzw. Überschriften, die die Phrase beinhalten. Wie in Abbildung 9 erkennbar besteht das Input – Layer aus vier Neuronen, die diese Merkmale als Input erhalten. Diese Inputsignale werden entsprechend den Gewichten der Verbindungskanten der einzelnen Layer abgeändert und am Output – Layer zusammengeführt. Der Wert des Neurons am Ausgang ist ein Maß dafür, ob die Phrase eine Schlüsselphrase ist oder nicht. Die Gewichte der Kanten werden dabei während einer Trainingsphase mit Dokumenten und deren Schlüsselwörtern geeignet adaptiert. Die von den Autoren durchgeführten Untersuchungen zeigen, dass ca. 65 % der von diesem System extrahierten Schlüsselwörter von Testpersonen als gut bewertet wurden und ca. 20 % als ungenügend (vgl. Wang et al., 2006).

4.2 Automatische Zusammenfassung von Texten

Die automatische Zusammenfassung von Texten (vgl. Abschnitt 3.3) ist dem Thema dieser Arbeit in vielerlei Hinsicht sehr ähnlich. Für ein funktionieren beider Techniken wird ein grundsätzliches semantisches Verstehen eines Textes vorausgesetzt, um die wesentlichen Inhalte aus diesen extrahieren zu können. Dies ist auch für die Automatische Fragengenerierung von essentieller Bedeutung, da die Fragen so gestaltet werden sollen, sodass damit überprüft werden kann, ob der wesentliche Inhalt des Textes verstanden worden ist. Aus diesem Grund werden nun nachfolgend einige Forschungsansätze der automatischen Textzusammenfassung präsentiert.

4.2.1 Klassische Ansätze

Der klassische Ansatz der automatischen Zusammenfassung stammt von Luhn (1958) und wurde bereits in Abschnitt 2.2.2 näher erläutert. Der Autor Edmundson (1969) erkannte die Probleme, die beim Ansatz von Luhn auftreten und entwickelte

seinerseits ein System, das bis heute starken Einfluss auf das Gebiet der automatischen Zusammenfassung ausübt (vgl. Mani, 2001).

Das System von Edmunson beruht auf vier grundsätzlichen Prinzipien. Diese sind (vgl. Edmundson, 1969):

1. *Cue Method*: Bei der Cue Method haben spezielle Wörter, die sogenannten Cue Phrases, ein vorher definierten Auswirkung auf das Gewicht eines Satzes. Solche Wörter sind beispielsweise wichtig, unmöglich, kaum usw. Dabei werden diese Wörter in drei Kategorien eingeteilt, nämlich Bonus - Wörter, Stigma - Wörter und Null - Wörter. Null - Wörter sind Präpositionen, Artikel, Pronomen, aber auch Verben wie beispielsweise sein. Diese Null - Wörter haben auf das Gewicht des Satzes keinen Einfluss. Stigma - Wörter, wie beispielsweise anaphorische Ausdrücke, verharmlosende Ausdrücke usw. sind Wörter, die das Satzgewicht negativ beeinflussen, wohingegen die Bonus - Wörter auf dieses Gewicht einen positiven Einfluss haben. Bonus - Wörter sind beispielsweise Vergleichsformen, Steigerungsformen, Schlussfolgerungen usw. Um das Satzgewicht für die Cue Method zu erhalten, werden die einzelnen Gewichte der Cue Phrases in einem Satz einfach addiert.

2. *Key Method*: Hierbei wird ähnlich der Methode von Luhn (1958) anhand der Häufigkeit des Auftretens von Wörtern in einem Dokument Schlüsselwörter (engl. keywords) extrahiert. Cue Phrases dürfen dabei allerdings nicht als Schlüsselwörter herangezogen werden. Alle Wörter, deren Häufigkeit über einem vorher definierten, von der Textlänge und Textart abhängigen, Schwellenwert liegt, werden als Schlüsselwörter berücksichtigt. Als Gewicht wird dabei die Häufigkeit der Wörter verwendet. Das Satzgewicht für die Key Method errechnet sich wiederum aus der Summe der einzelnen Gewichte innerhalb eines Satzes.

3. *Title Method*: Wörter, die im Titel oder in Überschriften vorkommen, sind üblicherweise für den Text bzw. für den Inhalt des Textes von großer Bedeutung. Daher werden jene Wörter, die in Überschriften oder im Titel auftreten und keine Null – Wörter sind, mit dem Text verglichen. Kommt nun eines dieser Wörter im Text vor, dann bekommt dieses ein höheres Gewicht zugeteilt. Das Endgewicht eines Satzes dieser Methode berechnet sich ebenfalls aus der Summe der Gewichte der Wörter innerhalb von diesem Satz.

4. *Location Method*: Diese Methode basiert auf zwei Annahme. Einerseits sind Sätze nach gewissen Überschriften von größerer Relevanz, andererseits sind Sätze, die Rückschlüsse auf das Thema zulassen, mit hoher Wahrscheinlichkeit am Beginn oder am Ende eines Dokumentes oder

Absatzes. Edmundson schlägt vor, das Sätze unter solchen speziellen Überschriften, wie beispielsweise Einleitung, Zusammenfassung, Zweck usw. positiver gewichtet werden. Außerdem bekommen Sätze auf Grund ihrer Position im Text, wie beispielsweise ersten und die letzten Sätze von Absätzen, ein höheres Gewicht. Das Satzgewicht ergibt sich wiederum Aus der Summe der einzelnen Gewichte eines Satzes.

$$W(s) = \propto C(s) + \beta K(s) + \gamma L(s) + \delta T(s) \qquad 4.4$$

Die Endgewichte der Sätze berechnen sich nach der Formel 4.4. Dabei bezeichnet die Parameter $C(s)$ das Gewicht, das mittels der Cue Method für den Satz mit der Nummer s berechnet wird, $K(s)$ jenes mit der Location Method berechnete Gewicht, $L(s)$ das Gewicht der Location Method und $T(s)$ Gewichte, das durch die Title Method ermittelt wird. Die Parameter $\propto$, β, γ und δ sind freie Parameter, die Edmundson (1969) nach Vergleich mit manuell erstellten Zusammenfassungen entsprechend wählte. Diese Parameter dienen dazu, den Einfluss der einzelnen der vier Methoden individuell zu gestalten und geben somit auch die Möglichkeit, auf verschiedene Textarten gesondert einzugehen (vgl. Mani, 2001).

Der Algorithmus zur automatischen Textzusammenfassung von Edmundson ist relativ einfach, dennoch haben sich einige seiner Erkenntnisse durchaus als zielführend herausgestellt. Der Autor Mani (2001) erläutert allerdings einige Kritikpunkte zu diesem Ansatz. So werden nur einzelne Elemente berücksichtigt und keine Sequenzen von Elementen. Darüber hinaus werden nur sehr oberflächliche morphologische und semantische Analysen durchgeführt und eine nicht-lineare Gewichtsberechnung könnte durchaus einige Vorteile mit sich bringen. Desweitern wird die Anzahl der zu extrahierenden Sätze nicht in die Berechnung mit einbezogen (vgl. Mani, 2001).

4.2.2 A Trainable Document Summarizer

Diese von den Autoren Kupiec, Pedersen und Chen (1995) vorgestellte Methode lernt anhand einer Menge von Dokumenten und den zu diesen Dokumenten gehörenden Zusammenfassungen welche Sätze aus einem Dokument extrahiert werden sollen. Dabei wird der Fokus auf fünf verschiedene Merkmale gelegt, die ein Indiz für die Wichtigkeit der Sätze darstellen sollen. Diese Merkmale beruhen teilweise auf den von Edmundson (1969) vorgestellten Verfahren, wobei diese erweitert bzw. adaptiert werden (vgl. Kupiec et al., 1995; Endress-Niggemeyer, 2001).

Diese fünf Merkmale sind (vgl. Kupiec et al., 1995; Endress-Niggemeyer, 2001):

1. *Satzlänge*: Kurze Sätze werden selten in Zusammenfassungen inkludiert. Daher ist es sinnvoll, einen Schwellenwert festzulegen, der Sätze, die mehr Wörter als dieser Schwellwert beinhalten, den Wert *true* zuweist, bei weniger Wörtern wird der Wert *false zugewiesen*.

2. *Indikatorphrasen*: Sätzen, die eine von 26 vordefinierten Phrasen wie beispielsweise *zusammenfassend* beinhalten wird *true* zugewiesen. Darüber hinaus werden Sätze, die direkt nach einer Überschrift kommen, welche Phrasen und Wörter wie Zusammenfassung, Ergebnisse, Diskussion usw. enthalten, mit *true* markiert.

3. *Absatzstruktur*: Dabei werden jeweils die ersten zehn und die letzten fünf Absätze eines Dokuments betrachtet. In den jeweiligen Absätzen werden die Sätze danach beurteilt, ob sie den Absatz einleiten, ihn abschließen oder ob sie sich in der Mitte des Absatzes befinden.

4. *Schlüsselwörter*: Die am häufigsten vorkommenden Inhaltswörter werden als Schlüsselwörter definiert. Jene Sätze, in denen die meisten Schlüsselwörter vorkommen werden mit *true* bewertet.

5. Eigennamen bzw. Akronyme: Eigennamen und deren Akronyme (z.B.: IBM) sind potentiell wichtige Wörter. Der Vorteil der englischen Sprache ist, dass im Gegensatz zur deutschen Sprache auch die Hauptwörter mit Kleinbuchstaben beginnen, es sei denn die Hauptwörter sind Eigennamen. Diese Tatsache nutzen Kupiec et al. (1995) aus, um Sätze, die Wörter bzw. Wortsequenzen beinhalten, welche mit einem Großbuchstaben beginnen, positiv zu bewerten. Dabei gelten allerdings die Restriktionen, dass dieses Wort kein Satzbeginn sein darf, dass dieses Wort keine Maßeinheit ist (z. B: F für Fahrenheit) und dass dieses Wort beziehungsweise diese Wortsequenz öfters als einmal im Text vorkommt. Des Weiteren wird das erste Vorkommen doppelt so hoch bewertet wie alle weiteren Vorkommen.

$$P(s \in S|F_1, F_2, \dots F_k) = \frac{\prod_{j=1}^{k} P\left(F_j \middle| s \in S\right) P(s \in S)}{\prod_{j=1}^{k} P(F_j)} \qquad 4.5$$

Mit Hilfe von Formel 4.5 berechnen Kupiec et al. (1995) nun die Wahrscheinlichkeit für jeden Satz dahingehend, ob dieser von einem Menschen in eine Zusammenfassung aufgenommen werden würde. Der Parameter *Fj* entspricht den einzelnen Merkmalen und *s* entspricht einem Satz. Dabei ist der Ausdruck $P(s \in S)$ konstant und wird laut Endress-Niggemeyer (2001) als Verkürzungsrate vorgegeben. Die Ausdrücke $P\left(F_j \middle| s \in S\right)$ und $P(F_j)$ können direkt aus dem Trainingsset ermittelt werden, indem man deren Häufigkeit ermittelt. Diese

Ausdrücke repräsentieren die Wahrscheinlichkeiten der Merkmale der Sätze in der Zusammenfassung beziehungsweise die Häufigkeit der Merkmale in einem Trainingsset (vgl. Kupiec et al., 1995; Endress-Niggemeyer, 2001).

Der trainierbare Zusammenfasser von Kupiec et al. (1995) erreichte eine Übereinstimmung von 35 % mit den von Menschen erzeugten Zusammenfassungen. Dabei sei allerdings erwähnt, dass selbst bei verschiedenen menschlichen Zusammenfassseren oft nur 25 % der Sätze übereinstimmen und dass selbst ein und derselbe Zusammenfasser bei größeren Zeitabständen zwischen dem Verfassen der Zusammenfassungen auch nur ungefähr 55 % an Übereinstimmung erreicht (vgl. Kupiec et al., 1995).

4.2.3 Maximal Frequent Sequences

Die Autoren Ledeneva, Gelbukh und García - Hernández (2008) erläutern in ihrer Arbeit *Terms Derived from Frequent Sequences for Extractive Text Summarization* den Einsatz von sogenannten *Maximal Frequent Sequences* (MFSs) für eine extraktive automatische Textzusammenfassung. MFSs sind n - Gramme von mindestens zwei Wörtern, die häufiger als ein vordefinierter Schwellwert zusammen in einem Text in der gleichen Reihenfolge vorkommen. Das Grundprinzip bei dieser Methode ist, dass solche n-Gramme sehr häufig wichtige semantische Informationen beinhalten und somit für eine Beschreibung des Inhalts essentiell wichtig sind (vgl. Ledeneva et al., 2008).

In ihren Experimenten verwenden die Autoren Ledeneva et al. (2008) den folgenden Algorithmus um herauszufinden, mit welchen Optionen die besten Ergebnisse erzielt werden.

Algorithmus (vgl. Ledeneva et al., 2008):

1. *Auswahl der Terme*: Bei der Auswahl der Terme untersuchen die Autoren Ledeneva et al. (2008) die Ergebnisse bei Auswahl von MFS (mit einer Mindestlänge von zwei), sowie Bi - Gramme (Wortsequenzen bestehend aus zwei Wörtern) und auch einzelne Wörter. Es zeigt sich dabei, dass die Zusammenfassungen bei Verwendung von MFSs die besten Ergebnisse liefern.

2. *Gewichtung der Terme*: Für die Gewichtung der Terme verwenden Ledeneva et al. (2008) drei verschieden Möglichkeiten. Einerseits werden die Terme dahingehend gewichtet, wie oft sie in MFCs im gesamten Text vorkommen. Andererseits wird einem Term als Gewicht die maximale Länge eines MFCs, welches diesen Term enthält, zugeteilt. Bei der dritten Möglichkeit wird allen Termen dasselbe Gewicht zugewiesen.

3. *Satzgewichtung*: Um das Gewicht für die einzelnen Sätze zu berechnen, werden alle Gewichte der Terme innerhalb eines Satzes addiert.

4. *Satzauswahl*: Bei der Satzauswahl werden zwei verschiedene Möglichkeiten verwendet. Einerseits werden die Sätze mit dem größten Gewichten ausgewählt, solange bis eine gewünschte Größe der Zusammenfassung erreicht wird. Andererseits wird eine vorher definierte Anzahl der besten Sätze ausgewählt, danach werden die ersten Sätze des Textes ausgewählt, bis die gewünschte Länge der Zusammenfassung erreicht wird.

Die Analysen Ledeneva et al. (2008) zeigen, dass die besten Ergebnisse mit MFSs und Bi - Grammen erzielt werden. Die besten Ergebnisse in Bezug auf die Term Gewichtung werden damit erreicht, den Termen die Anzahl des Vorkommens in MFCs als Gewicht zuzuweisen. Für die Satzauswahl wird vorgeschlagen den am höchsten gewichteten Satz zu verwenden, und danach die ersten Sätze des Textes. Dies hat allerdings dadurch begründet, dass die Autoren bei diesen Tests als Testdokumente nur Zeitungsartikel verwenden. Zeitungsartikel haben die Eigenschaft, dass die wichtigsten Informationen mit sehr großer Wahrscheinlichkeit in den ersten Sätzen beinhaltet ist (vgl. Ledeneva et al., 2008).

4.2.4 Lexikalische Ketten

Die Autoren Erkan und Cicekli (2008) zeigen in ihrem *Paper Lexical Cohesion Based Topic Modeling for Summarization* eine Methode der automatischen Textzusammenfassung, welche einen tiefere semantische Analyse des Textes erfordert als die bisher vorgestellten Algorithmen. Dabei werden sogenannte lexikalische Ketten verwendet. Mit Hilfe solcher lexikalischer Ketten ist es möglich, die lexikalische Struktur von Texten abzubilden. Eine lexikalische Kette ist ein Graph, dessen Knoten die Bedeutungen der Wörter repräsentieren und dessen Kanten die semantischen Relationen dieser Wörter darstellen. Um solche Graphen aus Texten zu generieren, werden semantische Netze, wie beispielsweise WordNet, benötigt um damit die Wortbedeutungsdisambiguierung durchführen zu können (vgl. Erkan & Cicekli, 2008).

Der von den Autoren vorgeschlagene Algorithmus beinhaltet prinzipiell folgende Schritte (vgl. Erkan & Cicekli, 2008):

1. *Satzbestimmung und Wortartenerkennung*: Dabei werden die einzelnen Sätze des Textes bestimmt und eine Wortartendisambiguierung (vgl. Abschnitt 2.2.1.2) durchgeführt (vgl. Erkan & Cicekli, 2008).

2. *Nominalphrasenerkennung*: Die Autoren Erkan und Cicekli (2008) ermitteln mit Hilfe der in Punkt 1 gewonnen Erkenntnisse die Nominalphrasen im Text. Eine Nominalphrase zeichnet sich dadurch aus, dass sie normalerweise mit

einem Schlüsselwort (einem Substantiv) enden. Diese Wörter werden dann mit anderen Wörtern in der Umgebung dieses Wortes in Verbindung gebracht und so die Nominalphrase gebildet. Es ist allerdings auch möglich, dass das Wort alleine eine Nominalphrase bildet (vgl. Erkan & Cicekli, 2008).

3. *Kettenbildung*: Für die Bildung der lexikalischen Ketten verwenden die Autoren Erkan und Cicekli (2008) den Algorithmus von Galley und McKeown (2003). Galley und McKeown (2003) wiederum verwenden WordNet (vgl. Abschnitt 6.2) und dessen Struktur um die semantischen Relationen zwischen den Wörtern zu extrahieren. Dies funktioniert in drei Schritten. Als erstes wird mit Hilfe der Hyperonym/Hyponym - Struktur von WordNet eine Repräsentation aller möglichen Bedeutungen des Textes gebildet. Im zweiten Schritt folgt eine Wortbedeutungsdisambiguierung und der dritte Schritt ist die Bildung der Ketten. Dabei werden aus der Repräsentation aller möglichen Bedeutungen jene Verbindungen entfernt, die nicht der richtigen Bedeutung des Textes entsprechen (vgl. Galley & McKeown, 2003).

4. *Entfernung schwacher Ketten:* Schwache Ketten sind Ketten die nur aus einem Wort bestehen. Diese werden in diesem Schritt als erstes entfernt, da diese Probleme bei der Identifizierung bzw. Segmentierung der Themen verursachen können. Für die Entfernung weiterer Ketten, welche den Ansprüchen nicht genügen, verwenden die Autoren Erkan und Cicekli (2008) die von Barzilay und Elhadad (1999) vorgeschlagenen Formeln:

$$Score(Chain) = Length * Homogeneity \tag{4.6}$$

$$Homogeneity = 1 - \frac{\#DistinctMembers}{Length} \tag{4.7}$$

$$Score(Chain) > Average\ (Scores) + 2 * StandardDeviation(Scores) \tag{4.8}$$

Die Berechnung der Bewertung der Ketten wird in Formel 4.6 dargestellt, wobei *Length* die Anzahl der Knoten einer Kette darstellt und der Faktor *Homogeneity* nach Formel 4.7 berechnet wird. Der Ausdruck *Score(Chain)* in Formel 4.8 ist gleich der Gesamtbewertung der Kette. Dieser Wert muss größer sein muss als das Produkt der durchschnittlichen Bewertung mit der zweifachen Standardabweichung der Bewertungen, da die Kette sonst aus der möglichen Menge von Ketten eliminiert wird. Nach Barzilay und Elhadad (1999) korreliert Formel 4.8 mit der Bewertung durch Menschen (vgl. Erkan & Cicekli, 2008; Barzilay & Elhadad, 1999).

5. *Clustering der Ketten*: Das Clustering der Ketten funktioniert mit Hilfe von Statistiken über das gemeinsame Auftreten der Wörter bzw. Ketten in einem Satz. Die Autoren Erkan *und* Cicekli *(*2008*)* gehen dabei von der Annahme aus, dass zwei lexikalische Ketten, die öfters in einem Satz auftreten, in diesem Kontext eng miteinander verflochten sind. Für das Clustering wird im ersten Schritt für jede Kette ein Vektor, wobei die Länge dieses Vektors die Anzahl der Sätze im Text entspricht, gebildet. Die Werte der Elemente der Vektoren sind gleich der Anzahl der Vorkommen der lexikalischen Ketten in dem Satz, welcher das Element repräsentiert. Die einzelnen Cluster werden nun über ein Ähnlichkeitsmaß (Kosinus des Winkels der Vektoren) für diese Vektoren gebildet, das bedeutet, Vektoren die sich sehr ähnlich sind, werden dem selbe Cluster zuordnet. Dieses Clustering ist ein iterativer Vorgang, bei dem zu Beginn jede Kette einen eigenen Cluster repräsentiert. In jedem Iterationsschritt werden dann die ähnlichsten Cluster zu einem Cluster zusammengefasst, solange, bis die Ähnlichkeit der einzelnen Cluster einen gewissen Schwellwert unterschreitet (vgl. Erkan & Cicekli, 2008)

6. *Extraktion von Sätzen*: Die in einem Cluster enthaltenen Lexikalischen Ketten werden dann miteinander verglichen, um Sequenzen von Sätzen zu finden, welche lexikalische Ketten beinhalten. Solche Sequenzen haben die Eigenschaft, dass diese Segmente bzw. die Sätze der Segmente ein bestimmtes Thema zum Inhalt haben, da Menschen beim Verfassen der Texte die Eigenschaft haben, ein Thema mit einem Satz einzuleiten, um dieses dann in den folgenden Sätzen zu erläutern. Diese Sequenzen werden dann entsprechend Formel 4.9 gewichtet. Dabei ist s_i die Sequenz i, $S(s_i)$ ist das Gewicht dieser Sequenz, der Ausdruck $S(Cl_i)$ ist das durchschnittliche Gewicht der Ketten im Cluster i, SLC_i ist die Anzahl der lexikalischen Ketten, die in dieser Sequenz beginnen, PLC_i ist Anzahl der lexikalischen Ketten, die einen Knoten zur Sequenz beitragen, L_i ist die Anzahl der Sätze in der Sequenz und f^2 ist die gesamte Anzahl der Ketten in diesem Cluster (vgl. Erkan & Cicekli, 2008).

$$S(s_i) = S(Cl_i) * L_i * \frac{(1 + SLC_i) * PLC_i}{f^2} \qquad 4.9$$

   Nach der Berechnung der Gewichte der einzelnen Sequenzen werden für die Zusammenfassung die besten Sequenzen ausgewählt und von diesen jeweils der erste Satz in die Zusammenfassung aufgenommen. Die Sätze werden dabei in Abhängigkeit des Gewichtes jener Sequenz, welche den Sätzen zugewiesen sind, geordnet (vgl. Erkan & Cicekli, 2008).

Das von Erkan und Cicekli (2008) vorgestellt System der Verwendung lexikalischer Ketten zur automatischen Zusammenfassung erzielt durchaus

akzeptable Ergebnisse dahingehend, dass es bei der Evaluierung mittels DUC2004 zufriedenstellende Resultate liefert (vgl. Erkan & Cicekli, 2008).

4.2.5 Event-Based Summarization Using Time Features

Die Autoren Wu, Li, Lu und Wong (2007) stellen in ihrem Paper *Event-Based Summarization Using Time Features* eine Methode der automatischen Textzusammenfassung vor, die Events und Zeitpunkte in den Texten nutzt, um eine Zusammenfassung zu generieren.

Dabei nutzen sie die Tatsache aus, dass es möglich ist, mit Events Texte, vor allem Nachrichtenberichte, zu repräsentieren. Anhand der Beispiele, die Wu et al. (2007) nennen, ist es allerdings zu erkennen, dass diese Methode nicht nur bei Nachrichtenberichten, sondern bei vielen Textarten eingesetzt werden kann.

Wu et al. (2007) beschreiben eine Event als "Wer hat Was Wem Wann Wo getan". Ein Event besteht dabei aus einem Eventterm und Eventelementen, wobei die Eventterme die Aktion selbst repräsentieren, die Eventelement symbolisieren dabei Wörter und Bezeichnungen für Personen, Orte, Zeitpunkte usw.

Um diese Events aus einem Text extrahieren zu können, verwenden Wu et al. (2007) das GATE Analyse Tool (vgl. Abschnitt 6.1), insbesondere die Wortartenerkennung sowie die Eigennamenerkennung, die von diesem Tool zur Verfügung gestellt werden. Im Anschluss an diese Analysen werden aus dem Text die Eventterme extrahiert. Eventterme sind dabei Verben und Substantive, welche Aktionen und Handlungen repräsentieren. Um diese zu extrahieren, bedienen sich Wu et al. (2007) WordNet (vgl. Abschnitt 6.2). Es werden dabei nur Wörter als Eventterme verwendet, die ein Hyponym zu den *unique beginners* "event" und "action" darstellen. Des Weiteren ist zu beachten, dass Eventterme immer zwischen zwei Eventelementen oder zumindest in relativer Nähe zu einem Eventelement vorkommen müssen (vgl. Wu et al., 2007).

Nach der Extraktion dieser Events wird dann eine Zeitlinie gebildet, wobei Zeitpunkte einen Punkt mit dem Gewicht eins an diesem spezifischen Tag erhalten, und Zeitperioden der Dauer von N Tagen erhalten an jedem der Tage innerhalb dieser Zeitperiode einen Punkt, der allerdings nur mit dem Gewicht 1/N versehen wird (vgl. Abbildung 10). Hierbei ist zu erwähnen, dass für die Einheit der Zeitlinie ein Tag ausgewählt wird, diese Einheit kann allerdings adaptiert werden. Events, welchen kein exakter Zeitpunkt zugeordnet werden kann, werden dennoch eingeordnet, indem diesen der nächstgelegen Zeitpunkt zugewiesen wird. Dies basiert auf die Annahme, dass auch Menschen dazu tendieren, solch einem Event den nächstgelegenen Zeitpunkt zuzuweisen (vgl. Wu et al., 2007).

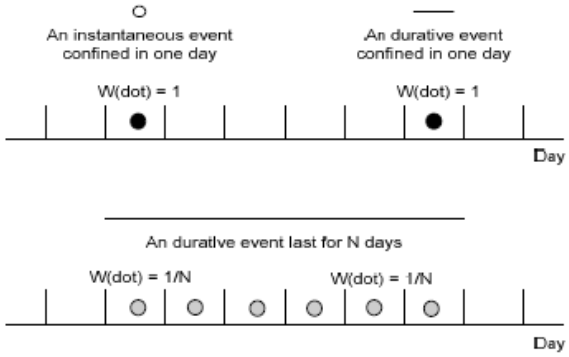

Abbildung 10: Beispielhafte Darstellung einer Zeitlinie (vgl. Wu et al., 2007)

Für einen Event, der vor oder nach einem bestimmten Zeitpunkt auftritt, werden nun vorher oder danach 7 Punkte in die Zeitlinie eingefügt. Für einen Event, der in einem Zeitraum um diesen Zeitpunkt stattfindet werden sowohl vorher als auch nachher 3 Punkte hinzugefügt. Die Gewichte der Punkte, die den Eventtermen bzw. den Eventelementen entsprechen, werden dabei mittels einer Normalverteilung ermittelt (vgl. Wu et al., 2007).

Mit Hilfe der Gewichte, die den Eventtermen bzw. den Eventelementen zugewiesen wurden, ist es nun möglich, die Sätze des Textes zu gewichten, indem alle Gewichte der Eventtermen bzw. den Eventelementen eines Satzes aufsummiert werden. Für die Satzauswahl gibt es zwei Möglichkeiten. Einerseits werden jene Sätze ausgewählt, die die höchsten Gewichte haben, anderseits kann für jeden Tag der Zeitlinie der jeweils beste Satz ausgewählt werden. (vgl. Wu et al., 2007)

4.2.6 Zusammenfassung durch Wortinformation und Satzposition

Die Autoren Cruz und Urrea (2005) stellen in ihrem Paper *Extractive Summarization on Word Information ans Sentence Position* eine Methode der extraktiven Zusammenfassung vor, die nur auf Worthäufigkeiten und Satzpositionen beruht. Ein Satz wird dabei nach Formel 4.10 gewichtet. Dabei ist n die Anzahl der Wörter in einem Satz und p_i ist die Worthäufigkeit des jeweiligen Wortes im ganzen Dokument. Der erste Faktor repräsentiert in dieser Formel die Wortinformation, der zweite Faktor repräsentiert die Satzposition, wobei o die Position des Satzes angibt und s die Anzahl der Sätze im Dokument. Durch diesen Term soll sichergestellt werden, dass Sätze, die am Ende eines Dokuments vorkommen, höher bewertet werden (vgl. Cruz & Urrea, 2005).

$$\frac{\sum_{i=1}^{n} \log_2(p_i)}{n} * \sqrt[25]{\frac{o}{s}} \qquad 4.10$$

Um die Performance zu erhöhen werden Wörter, die weniger als drei Mal im Text vorkommen, nicht berücksichtigt. Damit auch Funktionswörter die Effizienz des Algorithmus nicht beeinflussen, werden jene Worttypen verworfen, die weniger als die Hälfte der durchschnittlichen Wortinformation aller Worttypen im Dokument in sich tragen. Für die Satzauswahl werden nun die am höchsten bewerteten Sätze ermittelt und dann chronologisch in die Zusammenfassung eingefügt (vgl. Cruz & Urrea, 2005).

Bei den Evaluationen der Autoren Cruz und Urrea (2005), zeigten sich durchaus akzeptable Ergebnisse, obgleich diese Methode für verschiedene Textarten unterschiedliche Resultate liefert. So erbringt dieser Algorithmus bei technischen Berichten relativ schlechte Ergebnisse (vgl. Cruz & Urrea, 2005).

4.2.7 Semantic Summarization System

Das von den Autoren Bawakid und Oussalah (2008) vorgestellte System generiert anhand von statischen und dynamischen Merkmalen von Texten eine Zusammenfassung. Statische Merkmale sind dabei beispielsweise die Satzposition oder Eigennamen im Text, wohingegen dynamische Merkmale beispielsweise der semantischen Ähnlichkeit eines Satzes zu anderen Sätzen im Dokument entsprechen (vgl. Bawakid & Oussalah, 2008).

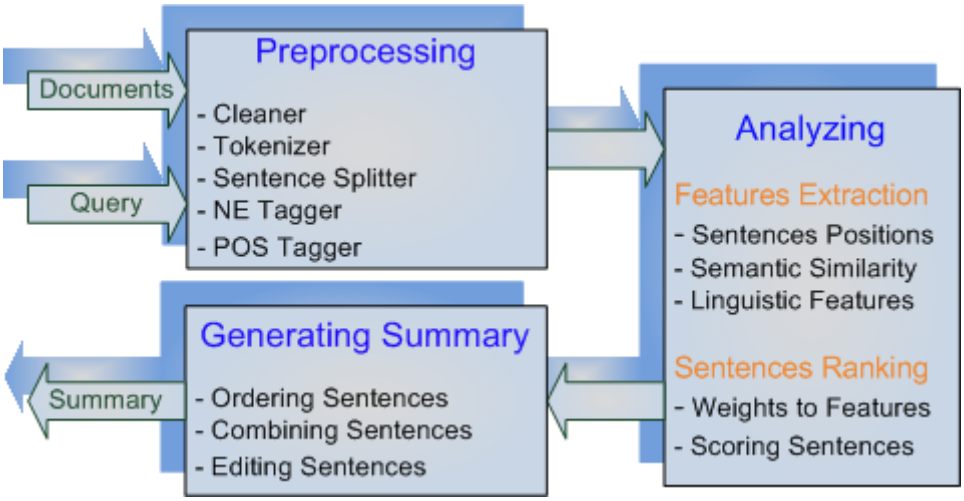

Abbildung 11: Architektur des Systems (vgl. Bawakid & Oussalah, 2008, überarbeitet)

Abbildung 11 zeigt einen Überblick über die prinzipielle Struktur des Systems. Die Query, also die Suchanfrage repräsentiert dabei das grundsätzliche Thema des Textes. Wie in Abbildung 11 erkennbar, gibt es bei diesem System drei grundsätzliche Abarbeitungsebenen, welche nachfolgend kurz erläutert werden (vgl. Bawakid & Oussalah, 2008).

In der Vorverarbeitungsebene wird zu Beginn ein sogenannter *Cleaner* verwendet, der die unwichtigen Zusatzinformationen eines Textes wie zum Beispiel HTML/XML - Tags, Verfasserinformationen usw. aus dem Ursprungsdokument entfernt. Mit Hilfe von GATE (vgl. Abschnitt 6.1) werden dann die weiteren Vorverarbeitungsschritte durchgeführt. Diese Schritte sind Satzgrenzenerkennung, Tokenisierung, Eigennamenerkennung sowie Wortartendisambiguierung (vgl. Bawakid & Oussalah, 2008).

In dieser Analyseebene werden einerseits die Merkmale extrahiert, andererseits werden die einzelnen Sätze gewichtet. Bei dieser Gewichtung werden Eigennamen, die Ähnlichkeit zum Titel beziehungsweise zur Query aufweisen berücksichtigt. Die Satzposition im Text wird ebenfalls bewertet. Sätze, die zu Beginn oder am Ende des Dokuments positioniert sind, werden dabei höher gewichtet. Darüber hinaus bekommen auch Sätze, die viele Eigennamen beinhalten oder eine große Ähnlichkeit zum Title oder zur Query aufweisen ebenfalls ein höheres Gewicht zugewiesen. Um die Ähnlichkeit von Sätzen zueinander zu bestimmen, ermitteln Bawakid und Oussalah (2008) zu Beginn für jedes Adjektiv das Substantive, welches dieses beschreibt und für jedes Adverb das Verb, welches dieses Adverb beschreibt. Drüber hinaus wird versucht, den Substantiven spezielle Pronomen aus dem Text zuzuweisen, die darauf hindeuten, dass dieses Wort wichtig ist. Solche Wörter sind beispielsweise viel, wenig, mehr usw. und werden mit Hilfe von vordefinierten Listen ermittelt (vgl. Bawakid & Oussalah, 2008).

Danach werden für jedes Substantiv und für jedes Verb in beiden Sätzen, die am höchsten gewichteten Übereinstimmungen im jeweils anderen Satz gesucht. Dabei wird einerseits die Ähnlichkeit der Wörter verwendet, andererseits wird ermittelt, ob die Adjektive bzw. die Adverbien der zu vergleichenden Wörter dieselben sind. Darüber hinaus werden auch die speziellen Pronomen dahingehend berücksichtigt, ob diese beiden Pronomen verstärkende oder abschwächende Aussagekraft haben oder ob sie entgegengesetzt wirken. Für die semantische Ähnlichkeit wird dabei der Algorithmus von Jiang und Conrath (1997) oder jener von Lin (1998) benutzt. Abschließend wird dann die semantische Ähnlichkeit zwischen diesen Sätzen ermittelt, indem die durchschnittliche Ähnlichkeit der Substantive und Verben beider Sätze gebildet wird. Das Ziel dieser Ähnlichkeitsberechnung ist es, alle Wörter beider Sätze in die Berechnung mit einzubeziehen (vgl. Bawakid & Oussalah, 2008).

Die Satzgewichtung wird anschließend anhand von Formel 4.11 vorgenommen.

$$Score\ (i) = \frac{(\propto * \mathrm{Sim}(s_i, T) + \beta * Sim(s_i, Q)) * n(s_i) * (F_{NE}(s_i) + 1) * P(s_i)}{N(NE + 1)} \qquad 4.11$$

Dabei ist N die Anzahl der Sätze im Dokument, $n(s_i)$ ist die Anzahl von Sätzen, die eine semantische Ähnlichkeit zum Satz s_i haben und zusätzlich größer

ist als ein vorher bestimmter Schwellwert. $P(s_i)$ ist das Gewicht, welches durch die Satzposition bestimmt wird, NE ist die Anzahl der Eigennamen im gesamten Dokument und $F_{NE}(s_i)$ ist gleich der Anzahl der Eigennamen im Satz i. $Sim(s_i, T)$ ist die Ähnlichkeit des Satzes i mit dem Titel des Textes und $Sim(s_i, Q)$ ist die Ähnlichkeit mit der Query. Die Summe Parameter $\propto$ und β muss eins betragen. Diese beiden Parameter sollen es ermöglichen, den Einfluss von Titel und Query individuell zu gestalten (vgl. Bawakid & Oussalah, 2008).

In der letzten Ebene wird die Zusammenfassung erzeugt. Dabei werden die am höchsten bewerteten Sätze ausgewählt und in chronologischer Reihenfolge in die Zusammenfassung eingefügt. Des Weiteren ist dieses System auch für Zusammenfassungen von mehreren Dokumenten verwendbar. Dabei werden die Dokumente einzeln evaluiert und danach die am höchsten bewerteten Sätze aller Dokumente in die Zusammenfassung integriert (vgl. Bawakid & Oussalah, 2008).

4.3 Extraktion von Konzepten und Ontologien

In diesem Abschnitt werden Methoden zur Extraktion von Konzepten, welche eine Repräsentation der semantischen Inhalte eines Textes darstellen, vorgestellt. Mit Hilfe dieser Konzepte können als die grundsätzlichen Themen und Ideen, welche hinter einem Text stehen, ausgedrückt werden.

4.3.1 Konzeptextraktion aus unstrukturierten Texten

Die Autoren Gelfand, Wulfekuhler und Punch (1998) stellen in ihrem Paper *Automated Concept Extraction from Plain Text* ein System zur Auffindung von semantischen Konzepten in unstrukturierten Texten vor. Sie verwenden dafür sogenannte *Semantic Relationship Graphs* (SRG). Diese Graphen stellen semantische Verbindungen zwischen den einzelnen Worten her und fügen, falls nötig, auch Verbindungswörter in den Graphen ein, die selbst nicht im Text vorkommen (vgl. Gelfand et al., 1998).

Die Ermittlung eines SRG erfolgt anhand eines rekursiven Algorithmus. Bei der Initialisierung wird als erstes eine Liste mit allen Wörtern im Text angelegt. Diese Wörter werden Basiswörter genannt. Im nächsten Schritt werden dann zu jedem Wort jene Basiswörter in der Liste gesucht, die eine direkte Verbindung zu diesem Wort haben. Diese Relationen werden mit Hilfe von WordNet (vgl. Abschnitt 6.2) bestimmt. Dieser Schritt wird dann für Wörter, die in Verbindung mit dem Ausgangswort stehen rekursiv weiterverfolgt, das bedeutet, es werden für die zugehörigen Wörter wiederum deren ähnliche Wörter aus den Basiswörtern gesucht und miteinander verbunden. Dies wird so lange fortgesetzt, bis eine gewisse Rekursionstiefe erreicht wird (vgl. Gelfand et al., 1998).

In SRGs, die auf diese Art und Weise gebildet werden finden Wörter, die keine semantische Verbindung zu einem anderen Wort, keine Berücksichtigung. Im Gegensatz dazu, werden Wörter, die zwei Basiswörter miteinander verbinden, aber selbst nicht Teil der Basiswörter sind, während der einzelnen Rekursionsschritte in den Graphen eingebunden. Wörter, die nur eine sehr schwache Verbindung zum Graphen aufweisen werden in weiterer Folge verworfen. Abbildung 12 zeigt einen Teil eines beispielhaften SRGs (vgl. Gelfand et al., 1998).

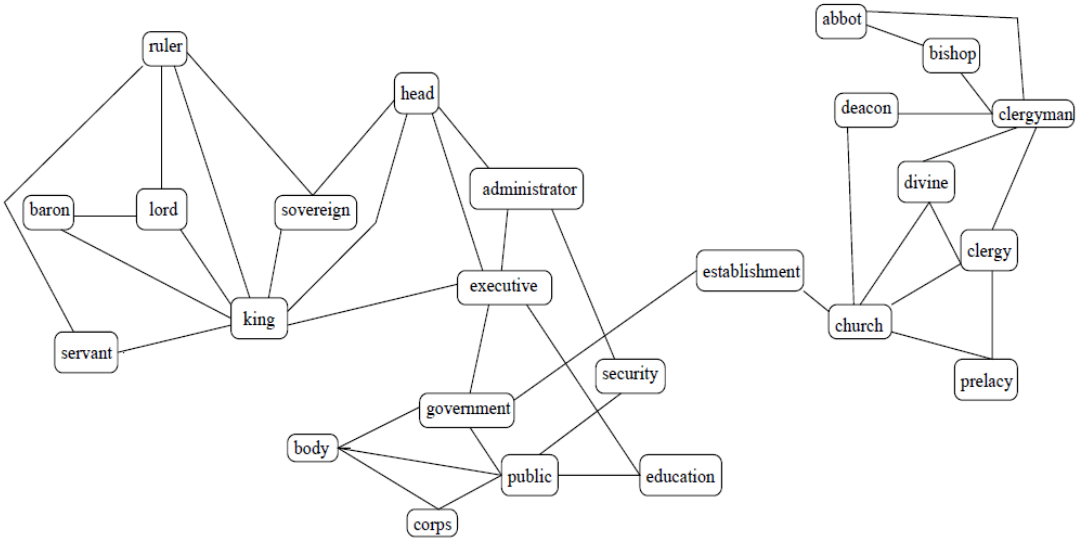

Abbildung 12: Beispiel eines SRG (vgl. Gelfand et al., 1998)

Wie in Abbildung 12 erkenntlich, sind *government* und *church* zwei zentrale Begriffe dieses Beispiels. Die zentralen Themen des Ausgangstextes sind die Reformbewegung in der Kirche sowie die Verbindungen dieser zu der Regierung (Königreich). Bemerkenswert sind die relativ starken Bindungen um diese zentralen Begriffe, obwohl bei diesem Beispiel beim der Erzeugung des Graphen nur mit eine Rekursionstiefe von zwei gearbeitet wurde (vgl. Gelfand et al., 1998).

Um nun die zentralen Themen dieser Graphen herauszufinden, werden von Gelfand et al. (1998) leicht modifizierte Single Link Clustering Algorithmen verwendet. Single Link Algorithmen sind Clustering - Verfahren, bei denen einem Cluster in jedem Schritt jenes Element zugewiesen wird, dass diesem am nächsten gelegen ist (vgl. Liu, 2009). Diese Clustering Algorithmen werden bei diesem System auf sogenannte Adjazenzmatrizen angewandt. Adjazenzmatrizen sind Matrizen, die den Graphen vollständig in einer Matrizenform abbilden (vgl. Cormen, Leiserson, Rivest und Stein, 2001). Gelfand et al. quadrieren bei ihrem Algorithmus die Adjazenzmatrix wodurch auch die Anzahl der Verbindungen zwischen zwei Wörtern stärker ins Gewicht mit einfließen. Dadurch ist es dann möglich, die einzelnen Subgraphen, welche die einzelnen Konzepte repräsentieren, zu extrahieren. (vgl. Gelfand et al., 1998).

4.3.2 Automatic Concept Extraction Algorithmen

Der von Ramirez und Mattmann (2004) vorgestellte Algorithmus zur automatischen Konzeptextraktion (ACE) wurde entwickelt, um die Ergebnisse von Internetsuchmaschinen zu verbessern. Die Verbesserung soll erreicht werden, indem bei der Suche auch die Übereinstimmung von Konzepten der Suchanfrage mit den Konzepten der Webseiten berücksichtigt werden. Aus diesem Grund wurde ACE hauptsächlich für HTML-Seiten entwickelt (vgl. Ramirez & Mattmann. 2004).

Der ACE - Algorithmus verarbeitet drei grundlegende Prinzipien. Einerseits wird die Worthäufigkeit verwendet, andererseits werden spezielle stilistische Informationen aus den HTML - Code extrahiert und in die Gewichtung der einzelnen Wörter mit einbezogen. Darüber hinaus werden auch Phrasen berücksichtigt. Die aus diesen drei Prinzipien gewonnen Gewichte werden kombiniert und jene Wörter und Phrasen, denen ein Gewicht zugeordnet ist, dass über einem vordefinierten Schwellwert liegt, werden ausgewählt und als Konzepte präsentiert (vgl. Ramirez & Mattmann. 2004).

4.3.3 Concept Extraction from student essays

Das von den Autoren Villalon und Calvo (2009) entwickelte System verwendet Grammatikbäume und den *Latent Semantic Analysis* Algorithmus (LSA) (vgl. Abschnitt 3.4.6) zur Konzeptextraktion. Mit Hilfe der Grammatikbäume werden die einzelnen Sätze eines Textes analysiert und die Phrasen und Substantive, die möglicherweise ein Konzept repräsentieren, extrahiert (vgl. Villalon & Calvo, 2009).

Um aus den möglichen Konzepten die geeigneten auszuwählen, wird LSA verwendet. Beim LSA werden für die Berechnung der Gewichte der einzelnen Wörter Worthäufigkeit und Dokumenthäufigkeit verwendet. Anschließend wird eine Matrix erzeugt, deren Element e_{ij} gleich dem Gewicht des Elements i im Textabschnitt j ist. Mittels einer Singulärwertzerlegung wird dann die Matrix in drei Matrizen zerlegt. Durch diese Zerlegung ist es dann möglich, die Dimension der Matrizen zu reduzieren und in weiterer Folge können aus der resultierenden Matrix die einzelnen Konzepte extrahiert werden, da jeder Eigenvektor dieser Matrix ein Konzept repräsentiert. Um die Anzahl der Konzepte zu verringern, wählen die Autoren Villalon und Calvo (2009) abschließend jene Wörter aus, welche zuvor auch als Substantive erkannt wurden und gleichzeitig ein hohes Gewicht haben (vgl. Villalon & Calvo, 2009).

4.3.4 Knowledge Based Topic Identification

Der Autor Lin (1995) entwickelte einen Prototyp zur automatischen Konzeptextraktion, welcher ohne morphologische Analysen arbeitet. Dafür definierte Lin (1995) die *concept frequency ratio* R (vgl. Formel 4.12), wobei C ein bestimmtes

Konzept, das bedeutet ein Wort oder eine Phrase, darstellt. Dieses Verhältnis gibt Aufschluss über den Informationsgehalt des Terms bezüglich einer Zusammenfassung. Ist dieser Wert sehr hoch, dann hat man eine relativ schlechte Generalisierung des Konzeptes ermittelt. Ist dieser Wert klein, so würde eine genauere Beschreibung des Konzepts einen größeren Informationsverlust bedeuten und ist daher nicht zielführend (vgl. Lin, 1995).

$$R = \frac{MAX(weigth\ of\ all\ the\ direct\ Children\ of\ C)}{SUM(weigth\ of\ all\ the\ direct\ Children\ of\ C)} \quad 4.12$$

Das in Formel 4.12 gezeigte Gewicht ($weigth$) ist der Worthäufigkeit gleichzusetzen. Um nun die interessanten Konzepte extrahieren zu können, wird ein Schwellenwert für *R* definiert. Jedes Wort bzw. jede Phrase ist genau dann als Konzept geeignet, wenn dafür ein Wert R berechnet wird, welcher unter diesem Schwellenwert liegt. Die dafür benötigte hierarchische Darstellung (vgl. Abbildung 13) der Wörter wird mit WordNet ermittelt (vgl. Lin, 1995).

Abbildung 13: Beispiel einer hierarchischen Darstellung (vgl. Lin, 1995)

4.3.5 Ontologieextraktion aus unstrukturiertem Text

Das von den Autoren Ahmad und Gillam (2005) entwickelte System zur Ontologieextraktion verwendet einen Korpus, welcher repräsentativ für eine gesamte Sprache ist, um die Unterschiede zu einem spezifischen Dokument einer bestimmten wissenschaftlichen Domäne herauszufinden. Aus diesen signifikanten Unterschieden der Häufigkeiten der einzelnen Wörter werden dann Konzepte und Ontologien für dieses Dokument extrahiert bzw. erzeugt (vgl. Ahmad & Gillam, 2005).

Im ersten Schritt bei diesem System zur Ontologieextraktion wird ein Korpus aus den zu untersuchenden Dokumenten gebildet und anschließend die Worthäufigkeiten ermittelt. Danach wird anhand der Formel 4.13 für jedes Wort im Dokument ein sogenannter *weirdness index* berechnet, der Aufschluss darüber gibt, wie häufig ein Wort im Dokument bzw. im repräsentativen Sprachkorpus vorkommt. Dabei sind f_{SL} und f_{GL} die Häufigkeit des Wortes im Dokumentkorpus bzw. im Sprachkorpus, N_{GL} und N_{SL} bezeichnen die jeweilige Größe dieser Korpera. Mit Hilfe

dieser Formel werden dann jene Wörter extrahiert, deren Häufigkeit im Dokument signifikant größer ist als im Sprachkorpus. Diese Wörter sind dann gut geeignet, die Konzepte in den Dokumenten zu beschreiben (vgl. Ahmad & Gillam, 2005).

$$weirdness = \frac{N_{GL} f_{SL}}{(1 + f_{GL}) N_{SL}} \quad 4.13$$

Die Worthierarchien werden dann mit Hilfe von Kollokationsanalysen erzeugt, wobei das gemeinsame Auftreten (Kollokation) einzelner Wörter im Dokument berücksichtigt wird. Die somit aus dem Dokument erzeugten Ontologien repräsentieren die Konzepte sowie die Verbindungen dieser Konzepte im Dokument (vgl. Ahmad & Gillam, 2005).

4.4 Zusammenfassung

In diesem Abschnitt wurde ein Überblick über den aktuellen Forschungsstand in einigen Bereichen des NLP gegeben. Dabei wurde der Fokus auf die Teilgebiete der Schlüsselwortextraktion, der automatischen Textzusammenfassung und auf die Konzept bzw. Ontologieextraktion gelegt, da diese Gebiete sehr eng mit dem Thema dieser Arbeit verflochten sind.

Es zeigte sich, dass viele dieser System auf den von Luhn (1958) und Edmundson (1969) vorgestellten Ansätzen beruhen. Dabei sind vor allem die Methoden der Worthäufigkeit und der Stoppwortentfernung zu nennen, die in sehr vielen Systemen zum Einsatz kommen. Darüber hinaus werden häufig syntaktischen Eigenschaften und strukturellen Merkmale der Texte mit einbezogen. Solche Eigenschaften und Merkmale sind beispielsweise die Position der Wörter bzw. der Sätze, das verwenden von Cue – Phrases, allgemein die Betrachtung von Phrasen anstatt von Wörtern, das mit einbeziehen von Titel und Überschriften sowie die Grundformreduktion. All diese Methoden können prinzipiell miteinander kombiniert werden, um die wichtigsten Phrasen und Wörter in den Texten zu identifizieren.

Verbesserungen dieser Methoden werden dann in weiterer Folge durch semantische Analysen sowie durch den Einsatz von maschinellem Lernen erzielt. Die semantischen Analysen werden oftmals mit WordNet durchgeführt, allerdings wird auch der Ansatz der Kollokation häufig eingesetzt. Bei den tieferen semantischen Analyseverfahren werden meistens semantische und lexikalische Verbindungen zwischen Wörtern hergestellt und daraus eine Beschreibung der Konzepte extrahiert und Ontologien generiert.

Maschinelles Lernen bringt durchaus viele Vorteile mit sich, hat aber einen großen Nachteil, da dabei stets Trainingsmengen für den Lernprozess dieser

Algorithmen vorhanden sein müssen. Das Erstellen dieser Trainingsmengen muss von Hand geschehen und ist im Normallfall sehr aufwendig. Diese Tatsache begrenzt natürlich die Einsatzmöglichkeiten des maschinellen Lernens in Hinblick auf das Thema dieser Arbeit.

Anschließend werden im nächsten Kapitel die aus den vorangegangenen Kapiteln gewonnen Erkenntnisse genutzt, um Anforderungen und Konzepte für ein System zur automatischen Extraktion von semantisch relevanten Daten zu definieren und darzustellen.

5 Automatische Extraktion von semantisch relevanten Daten

In diesem Kapitel werden die Erkenntnisse aus den vorangegangenen Kapiteln aufgegriffen, um Anforderungen an ein System für die Extraktion von semantisch relevanten Daten aufzuzeigen und Konzepte zu entwickeln, welche es ermöglichen, diesen Anforderungen gerecht zu werden.

5.1 Anforderungen

Das grundlegende Ziel des gesamten Systems ist es, semantisch relevante Daten aus natürlich sprachlichen Texten zu extrahieren um mit Hilfe dieser Daten (Wörter, Phrasen und Sätze) eine automatische Fragengenerierung zu ermöglichen. Diese Arbeit fokussiert sich dabei auf die Extraktion der Daten. Für die Umsetzung der Fragengenerierung mittels dieser extrahierten Daten sei auf Lankmayr (2010) verwiesen. Die extrahierten Daten sollen in diesem Fall den grundsätzlichen Inhalt, sowie die Ideen und Konzepte im Text repräsentieren, um in weiterer Folge mit den erstellten Fragen die Möglichkeit einer Überprüfung des Verständnisses des Textes bieten zu können.

Um diese Konzepte aus den Dokumenten extrahieren zu können soll das System spezielle Merkmale und Eigenschaften der Wörter in den Texten bzw. Merkmale und Eigenschaften der Dokumente selbst identifizieren. Diese Merkmale sollen anschließend in eine geeignete (numerische) Repräsentation transformiert werden, welche dann in weiterer Folge als Gewicht eines Wortes bezeichnet wird. Das Gewicht bietet in diesem System die Möglichkeit, die große Anzahl der Merkmale und Eigenschaften, die bei diesem System extrahierten werden, miteinander zu kombinieren, um so die Identifizierung der wesentlichen inhaltlichen Konzepte in den Dokumenten effizient gestalten zu können.

Des Weiteren soll das System Möglichkeiten bieten, in den Prozess der Extraktion der Konzepte einzugreifen. Dabei soll es das System ermöglichen, bestimmte Merkmale im Text bevorzugt (bzw. auch nachteilig) zu behandeln, um so die Einflüsse dieser Merkmale während des Auswahlverfahrens entsprechend zu gestalten. Dies soll durch individuelle Parametereinstellungen durchführbar sein, welche die Wichtigkeit dieser Merkmale, wie beispielsweise bestimmte Wort- und Phrasentypen, oder auch die Relevanz von Überschriften, Titel etc., repräsentieren.

Prinzipiell soll das System universell einsetzbar sein, das bedeutet, es soll möglichst viele Textformate unterstützen, sowie die Möglichkeit bieten, auch URLs anzugeben und den dahinterstehenden textuellen Inhalt dieser Internetadresse als Eingangsdokument zu nutzen. Dabei sollen den Texten auch keine Restriktionen hinsichtlich der Form und der Struktur auferlegt werden. Zusätzlich soll das

Programm für alle Textarten aus den unterschiedlichsten wissenschaftlichen Domänen Unterstützung bieten, ohne dass dem System dabei zusätzliches Wissen zugeführt werden muss.

5.2 Konzeptionelles Design

In diesem Abschnitt werden die prinzipiellen Konzepte, welche in weiterer Folge für die Umsetzung des Systems verwendet werden, aufgezeigt. Die Konzepte gliedern sich im Wesentlichen in vier Kategorien. Diese sind Vorverarbeitung, Textanalyse, Ermittlung der relevanten Daten und die abschließende Extraktion dieser Daten. Die grundlegende Architektur des Systems wird in Abbildung 14 illustriert.

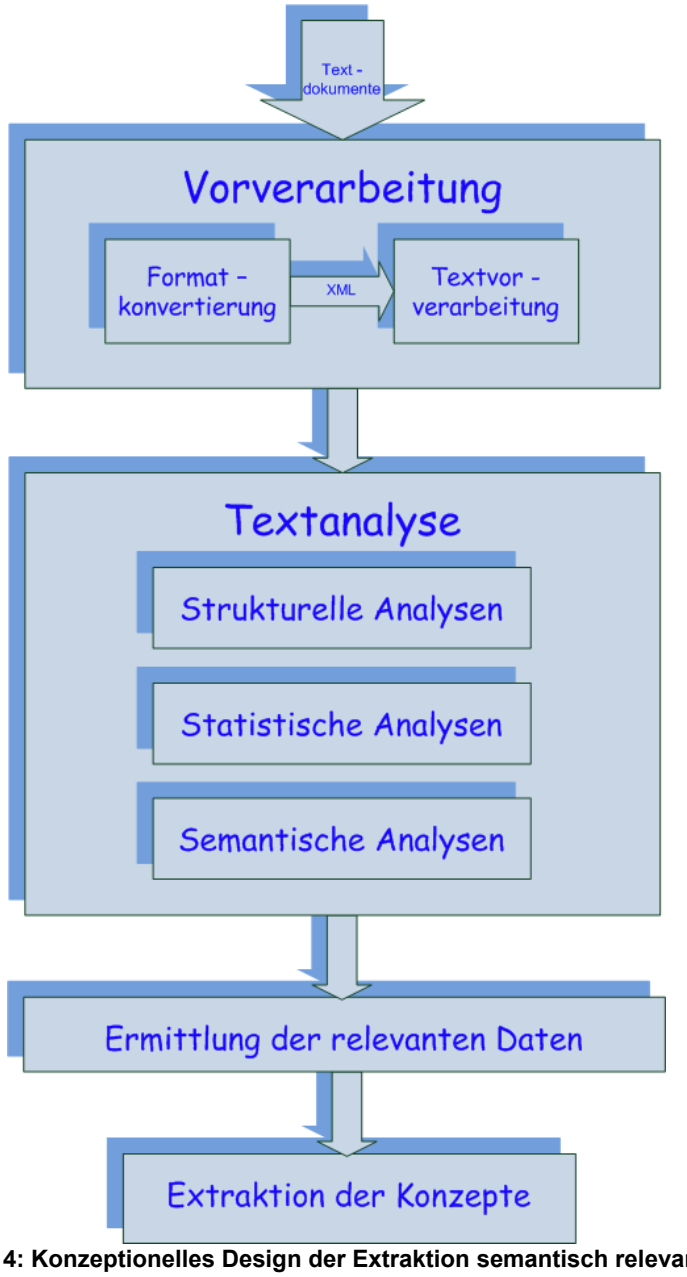

Abbildung 14: Konzeptionelles Design der Extraktion semantisch relevanter Daten

5.2.1 Vorverarbeitung

Bei der Vorverarbeitung wird einerseits eine Konvertierung der verschiedenen unterstützen Dateiformate in ein geeignetes Format umgesetzt, andererseits die Textvorverarbeitung durchgeführt. Der erste Schritt bei der Vorverarbeitung ist es, die verschiedensten Formate der Eingangstexte in ein einheitliches, standardisiertes Format umzuwandeln. Dabei muss aber verhindert werden, dass wichtige Informationen des Textes selbst bzw. der Struktur des Textes verloren gehen.

Die Textvorverarbeitung ist ein essentieller Schritt bei der Textverarbeitung. Je besser und umfangreicher dieser gestaltet wird, umso zielführender und effektiver wird die Durchführung der einzelnen darauf folgenden Verarbeitungsschritte. Deshalb kommt diesem Schritt eine wesentliche Bedeutung zu. Die Textvorverarbeitung besteht im Wesentlichen aus morphologischen und syntaktischen Analysen. Diese Analysen sind hauptsächlich die Wortartenerkennung sowie die Satzgrenzenerkennung. Darüber hinaus wird zusätzlich auch ein Stemming durchgeführt, um die Grundformen der einzelnen Wörter zu ermitteln.

Ein weiterer wichtiger Schritt bei der Vorverarbeitung ist die Eigennamenerkennung und das Chunking. Dabei werden spezielle Ausdrücke wie Namen von Personen, Orten, Unternehmen usw., sowie spezielle Phrasen wie Nominalphrasen, Verbalphrasen usw. in den zu untersuchenden Texten identifiziert. Wie aus Kapitel 4 ersichtlich, beinhalten diese dabei extrahierten Phrasen im Allgemeinen einen größeren Informationsgehalt als einzelne Wörter. Deshalb ist es wichtig, diese Daten in den Texten aufzufinden und dementsprechend zu behandeln.

5.2.2 Textanalyse

Die Analyse des Textes unterteilt sich in drei wesentliche Abschnitte. Einerseits werden die strukturellen Eigenschaften untersucht, andererseits finden statistische Analysen der Texte statt. Den Abschluss der Textanalyse bildet die semantische Analyse, welche teilweise auf die Erkenntnisse der statistischen und strukturellen Analysen aufbaut.

Strukturelle Analyse

Die strukturelle Analyse gliedert sich in drei Teilbereiche. Einerseits werden spezielle Strukturen in den Dokumenten untersucht, andererseits wird die Struktur des Dokuments selbst analysiert. Abschließend werden spezielle Formatierungen in den Texten behandelt. Diese Teilbereiche werden nun nachfolgend kurz erläutert.

1. *Spezielle Strukturen*: Spezielle Strukturen sind Wörter, Phrasen und Zeichen, die eine spezielle Bedeutung haben. Dies sind beispielsweise Datumsangaben, Adressen, E-Mail Adressen, URLs usw., aber auch Akronyme und anaphorische Ausdrücke. Diese speziellen Strukturen sind oftmals von großer Bedeutung für den Inhalt der Texte bzw. sind für die weiteren Analysen wichtig. Aus diesem Grund wird versucht, diese Terme in den Texten aufzufinden bzw. aufzulösen und dementsprechend zu markieren, um sie geeignet in den Analyseprozess integrieren zu können.

2. *Dokumentstruktur*: Einige der in Kapitel 4 vorgestellten Ansätze verwenden für ihre Analysen der Texte spezielle strukturelle Eigenschaften der Dokumente. Es zeigte sich, dass bestimmte Überschriften, wie zum Beispiel Kurzfassung (Abstract), Zusammenfassung, Schlüsselwörter (Keyphrases) usw. Hinweise für wichtige inhaltsrelevante Daten darstellen. Aus diesem Grund werden Wörter, die in den Textpassagen dieser Überschriften vorkommen als besonders wichtig behandelt. Des Weiteren gibt es Wörter und Phrasen, welche zwar für das gesamte Dokument eine relativ geringe (statistische) Aussagekraft haben, allerdings in einem kleineren Abschnitt des Dokuments eine wesentliche Bedeutung aufweisen. Solche Wörter können daher auch für die Bedeutung des gesamten Inhalts essentiell wichtig sein. Dieser Umstand beeinflusst die Analyse des Textes dahingehend, dass diese Analyse im Normalfall abschnittsweise durchgeführt wird, um so aus jedem Abschnitt die wichtigsten Konzepte extrahieren zu können. Die Abschnitte werden dabei durch die Überschriften der höchsten Ebene gegliedert.

3. *Spezielle Formatierungen*: Ein weiteres wichtiges Konzept stellt die Berücksichtigung von speziellen Formatierungen in einem Text dar. Dabei geht es vor allem um Terme, die **fett** oder *kursiv* formatiert, bzw. <u>unterstrichen</u> sind. Solche Wörter und Phrasen haben im Allgemeinen eine wichtige Bedeutung und sollten daher bei der Extraktion von semantisch relevanten Daten mit einbezogen werden.

Statistische Analyse

Die statistische Analyse ist hierbei dem Zählen der Vorkommen eines Wortes gleichzusetzen, da, wie bereits von Luhn (1958) erkannt, Inhaltswörter, welche häufig in einem Dokument vorkommen, mit sehr großer Wahrscheinlichkeit von essentieller Bedeutung für den Inhalt sind. Bei der Ermittlung der Häufigkeiten wird hierbei einerseits darauf geachtet, dass die Wörter auf ihre Grundformen reduziert sind, um die tatsächliche Anzahl der Wörter ermitteln zu können. Andererseits sollen auch Akronyme, Koreferenzen und Anaphorische Ausdrücke bei der Berechnung mit einbezogen werden, damit das Ergebnis möglichst exakt ist. Die hierbei ermittelten Häufigkeiten werden als zentraler Ausgangspunkt für die Ermittlung

eines Gesamtgewichtes für jedes Wort verwendet. Dieses Gewicht wird anschließend von den anderen in diesem Kapitel vorgestellten Konzepten, bzw. in weiterer Folge von den konkreten Umsetzungen dieser in der Applikation, in entsprechender Art und Weise adaptiert. Ziel ist dabei, es abschließend zu ermöglichen, jene Phrasen und Wörter zu extrahieren, welche die höchsten Gewichte aufweisen. Diese Phrasen sollen dann die wesentlichsten Kernkonzepte des Textes repräsentieren.

Semantische Analyse

Die Auflösung von semantischen Relationen zwischen den Wörtern in einem Text ist neben der statistischen Analyse das dritte Hauptkonzept des Systems zur Extraktion von inhaltsrelevanten Daten. Wie aus den vorangehenden Kapiteln ersichtlich, ist eine semantische Analyse der Texte eine Grundvoraussetzung für eine zufriedenstellende Funktionsweise eines solchen Systems.

Dieses Konzept kann dabei in weitere kleinere Teilkonzepte aufgeteilt werden. Diese Teile werden nachfolgend erläutert, wobei zu beachten ist, dass sich die einzelnen semantischen Relationen der Wörter in geeigneter Art und Weise auf das Gesamtgewicht auswirken.

1. *Wörter*: Für die Ermittlung der semantischen Verbindungen der Wörter in einem Text untereinander werden nur die Wörter der Textkörper verwendet. Dabei werden allerdings all jene Wörter, die zur Kurzfassung gehören, in diesem Schritt ignoriert, da diese gesondert behandelt werden. Bei diesem Konzept wird für jedes Wort ein Ähnlichkeitsmaß zu allen anderen Wörtern im selben Abschnitt gebildet. Zwei Wörter sind sich ähnlich, wenn das Ähnlichkeitsmaß über einen bestimmten definierbaren Wert liegt. Dies ist der Fall, wenn die Wörter in einer engen semantischen Beziehung zueinander stehen. Das dabei ermittelte Ähnlichkeitsmaß wirkt sich dann in weiterer Folge auf die Gewichte der Wörter aus.

   Dieses Konzept der semantischen Ähnlichkeit beruht auf der Annahme, dass ein zu einem wichtigen Wort in Verbindung stehendes Wort mit relativ hoher Wahrscheinlichkeit ebenfalls wichtig ist. Des Weiteren soll damit auch das Problem umgangen werden, welches bei der statistischen Analyse durch das Auftreten von Synonymen verursacht wird. Dabei kann es sein, dass bei häufiger Nutzung von Synonymen die Ergebnisse der statistischen Analyse keine Rückschlüsse auf die Wichtigkeit der Wörter zulassen. Durch diese Art der semantischen Analyse können diese Probleme aber teilweise vermieden werden.

2. *Überschriften*: Überschriften sind für die Bedeutung von Texten sehr wichtig, da sie oftmals einen direkten Hinweis auf den Inhalt der darauffolgenden

Textpassage geben. Deshalb werden Wörter, die in direkten Bezug zur Überschrift stehen positiver bewertet.

3. *Titel*: Ähnlich wie die Überschriften hat auch der Titel im Allgemeinen eine große Aussagekraft über den Inhalt des Textes. Der Titel soll normalerweise den Inhalt repräsentieren bzw. zumindest sehr eng mit dem Thema des Textes in Verbindung stehen. Deshalb werden Wörter, die mit dem Titel sehr ähnlich sind, ebenfalls positiver bewertet.

4. *Titel mit Überschriften*: Dabei werden jene Überschriften höher bewertet, welche mit dem Titel eine semantische Verbindung aufweisen. Das bedeutet, dass alle Wörter, die zu einer bestimmten Überschrift, welche zum Titel eine große semantische Ähnlichkeit aufweist, gehören, höher bewertet werden. Des Weiteren ist dadurch die Möglichkeit gegeben, Textpassagen niedriger zu bewerten, falls sie zum Titel und damit zum Thema des Textes nur eine geringe Relevanz aufweisen.

5. *Kategorien*: Dies beruht auf der Annahme, dass für unterschiedliche Kategorien von Texten unterschiedliche Strukturen und Merkmale der Texte für den Inhalt bedeutungstragend sind. Daher werden je nach Kategorie diese Merkmale bzw. die Wörter, die diesen Merkmalen entsprechen, höher bewertet. Solche Merkmale können beispielsweise spezielle Annotationen wie Datumsangaben, Namen usw. sein.

5.2.3 Ermittlung der relevanten Daten

Bei der Ermittlung der semantisch relevanten Daten werden die Ergebnisse der Textanalyse dazu verwendet, jene Daten (Wörter und Phrasen) aus den Texten zu extrahieren, die den Inhalt des Textes am besten widerspiegeln. Dabei werden die einzelnen Ergebnisse der Analysen in geeigneter Art und Weise kombiniert. In Abhängigkeit dieser Ergebnisse und der Parametereinstellungen wird ein Gesamtgewicht für alle Wörter im Text gebildet. Die relevanten Wörter und Phrasen können dann anhand der ermittelten Gewichte identifiziert werden.

5.2.4 Extraktion der Konzepte

Hierbei werden jene Wörter und Phrasen aus dem Text extrahiert, welche die wesentlichsten Konzepte in den Texten repräsentieren. Dabei werden jene Wörter, welche ein über einem definierbaren Schwellenwert liegendes Gesamtgewicht aufweisen, als relevant identifiziert. Mittels dieser Wörter werden anschließend geeignete Phrasen gesucht, anshcließend extrahiert und dem Benutzer in geeigneter Art und Weise zugänglich gemacht.

5.3 Zusammenfassung

In diesem Kapitel wurden die allgemeinen Anforderungen an ein System zur Extraktion von semantisch relevanten Daten aus natürlich sprachlichen Inhalten aufgezeigt sowie das konzeptionelle Design eines solchen Systems vorgestellt

Die grundsätzliche Anforderung an die extrahierten Daten ist dabei, dass diese Konzepte die wesentlichsten Inhalte der Dokumente repräsentieren sollen. Darüber hinaus soll es mittels dieser Konzepte möglich sein, eine automatische Fragengenerierung durchzuführen, wobei die erzeugten Fragen ebenfalls den wesentlichen Inhalt des Textes abdecken sollen. Eine weitere Anforderung an das System ist die Unterstützung von möglichst vielen Textarten und möglichst vieler Dateiformte. Zusätzlich soll das System unabhängig vom Wissensbereich des Textes sein, d.h. das System soll für jede beliebige (Wissens-) Domäne zufriedenstellende Ergebnisse liefern.

Das daraus resultierende konzeptionelle Design des Systems gliedert sich in vier wesentliche Teilbereiche, nämlich Vorverarbeitung, Textanalyse, Ermittlung der relevanten Daten sowie die Extraktion dieser Daten. Bei der Vorverarbeitung findet die Transformation der zu analysierenden Texte in ein geeignetes Format statt. Des Weiteren werden dabei grundlegende morphologische und syntaktische Analysen des Textes durchgeführt, um diesen für die weitere Verarbeitung vorzubereiten.

Die anschließende Textanalyse gelidert sich in die strukturelle, statistische und semantische Analyse. Der Prozess der Textanalyse dient dazu, besondere Eigenschaften und Merkmale in den Texten aufzufinden. Dabei werden hauptsächlich die Dokumentstruktur, die Anzahl des Vorkommens eines Wortes im Text, sowie die semantischen Ähnlichkeiten der Wörter untereinander untersucht. Diese Eigenschaften und Merkmale werden dann bei der Ermittlung der relevanten Daten dazu genutzt, die essentiellen Konzepte der Daten zu identifizieren.

Um die Umsetzung der in diesem Kapitel aufgezeigten Konzepte dieses Systems zu ermöglichen, werden einige bereits existierende Tools und Frameworks verwendet, da diese teilweise die benötigte Funktionalität in zufriedenstellender Art und Weise anbieten und somit eine eigene Umsetzung der Funktionalität nicht zielführend ist. Diese Tools und Frameworks werden im nachfolgenden Kapitel aufgezeigt und näher beschrieben.

6 Tools und Frameworks

In diesem Kapitel werden die Tools und Frameworks, welche in weiterer Folge bei der Implementierung des Systems benutzt werden, näher erläutert. Für eine genauere Beschreibung der Integration dieser Tools und Frameworks in das System und deren genaueren Aufgaben darin sei hier auf Kapitel 7 verwiesen.

6.1 Gate

A *General Architecture for Text Engineering* (GATE) ist eine Architektur, eine Entwicklungsumgebung und ein Framework für sprachwissenschaftliche Anwendungen. Architektur ist dabei als die Organisation und die Struktur des Systems, welche die einzelnen Komponenten untereinander koordiniert zu verstehen. Sie hat die Aufgabe die Kommunikation dieser Komponenten zu regeln und sicherzustellen, dass die Anforderungen, welche an das System gestellt werden, auch erfüllt werden. Die Funktion von GATE als Entwicklungsumgebung besteht darin, die Neuentwicklung von Systemen für sprachwissenschaftliche Analysen zu beschleunigen, indem es unterstützende Mechanismen für die allgemeine Entwicklung und für das Debugging bereitstellt. Als Framework bietet GATE eine Bibliothek mit einer Vielzahl von verschiedensten sprachwissenschaftlichen Ressourcen, sowie eine Schnittstelle für das Importieren und Exportieren von Daten, für das Einbinden der Ressourcen in Analyseverfahren und für die Darstellung der Daten bzw. der Ergebnisse der Analysen (vgl. Cunningham, 2002; Cunningham et al., 2002).

Die Ressourcen, welche GATE bietet, werden in sprachliche, sprachverarbeitende und darstellende (grafische Oberfläche) unterteilt. Sprachliche Ressourcen sind Methoden und Algorithmen, welche Korpora, Lexika, Ontologien erstellen und verwalten. Sprachverarbeitende Ressourcen sind jene, die für die sprachlichen Analysen verwendet werden wie beispielsweise Wortartenerkennung, Grundformreduktion, Koreferenzauflösung etc. All diese Ressourcen werden unter dem Begriff CREOLE (*Collection of REusable Objects for Language Engineering*) zusammengefasst (vgl. Cunningham, 2002; Cunningham et al., 2002).

Die grundsätzlichsten Ressourcen, die in sehr vielen NLP Verfahren zum Einsatz kommen, sind im ANNIE Plug-In (*A Nearly New IE System*) zusammengefasst, um die Verwendung dieser zu erleichtern. Dieses Plug-In beinhaltet einen Tokenisierer, eine Satzgrenzenerkennung, eine Wortartenerkennung, einen Eigennamenerkennung, eine Koreferenzauflösung, und einen semantischen Tagger. Dieser dient dazu, spezielle semantische Information, wie beispielsweise Datumsangaben und Adressen, aus dem Text zu

extrahieren und anschließend diese Informationen den entsprechenden Token zuzuordnen (vgl. Cunningham, 2002; Cunningham et al., 2002).

In GATE ist es prinzipiell auch möglich, die Ressourcen beliebig miteinander zu kombinieren, obgleich einige das vorhergehende Abarbeiten von anderen bedingen. Die aus den Prozessen gewonnen Informationen werden dem Benutzer dann mittels eines speziellen Annotierungsschemas zur Verfügung gestellt (vgl. Cunningham, 2002; Cunningham et al., 2002.; Cunningham et al., 2010).

Im Zuge dieser Arbeit wird auf eine detailliertere Beschreibung von GATE verzichtet. Für weitere Informationen sei auf Cunningham et al. (2010) beziehungsweise auf GATE (2010) verwiesen.

6.2 Wordnet

WordNet ist eine an der Princeton Universität entwickelte und von Hand erstellte lexikalische Datenbank der englische Sprache. In der dritten Version dieser Datenbank sind an die 160 000 Wörter (Substantive, Verben, Adjektive und Adverbien) gespeichert. Alle Wörter sind in Mengen von Synonymen (*synsets*) gruppiert, welche jeweils ein bestimmtes Konzept repräsentieren. Diese Synsets, ca. 120 000 in der dritten Version, sind untereinander mittels semantischer und lexikalischer Relationen verbunden. (vgl. WordNet, 2010; Fellbaum, 1998a).

Struktur von WordNet

In WordNet werden alle Wörter in verschiedene, hierarchisch angeordnete, Kategorien eingeteilt. Jede dieser Kategorien hat ein spezielles Wort (*unique beginner*) als Wurzel. Die unique beginner repräsentieren jeweils eine spezifische semantische Bedeutung bzw. Kategorie, wie beispielsweise Kommunikation, Tiere, Gefühle etc. Alle Wörter werden anhand ihrer Synonymitäten bzw. anhand der Hyperonym/Hyponym Struktur in die entsprechenden Hierarchien eingeordnet. Für Substantive gibt es 25 unique beginner, was der Anzahl der Dateien entspricht, in denen alle Substantive abgespeichert sind. Die Anzahl der unique beginner kann allerdings durch geschickte Gruppierung auf elf reduziert werden (vgl. Abbildung 15) (vgl. Miller, 1999).

Verben werden anhand von 15 unique beginner eingeordnet, welche sich allerdings von den unique beginner der Substantive unterscheiden. Eine Kategorie beinhaltet beispielsweise Zustandsverben, die anderen sind unterteilt in Verben, die Bewegungen, Kommunikation, Besitz etc. ausdrücken (Fellbaum, 1998b).

Adjektive werden in WordNet in zwei Klassen unterteilt. Einerseits gibt es beschreibende Adjektive (z. B.: groß, schön, schnell etc.), andererseits gibt es relationale Adjektive (z.B.: familiär), die aus Ableitungen von Substantiven

entstehen und meist eine sehr enge Beziehung zu diesem Wort aufweisen (vgl. Miller, 1999).

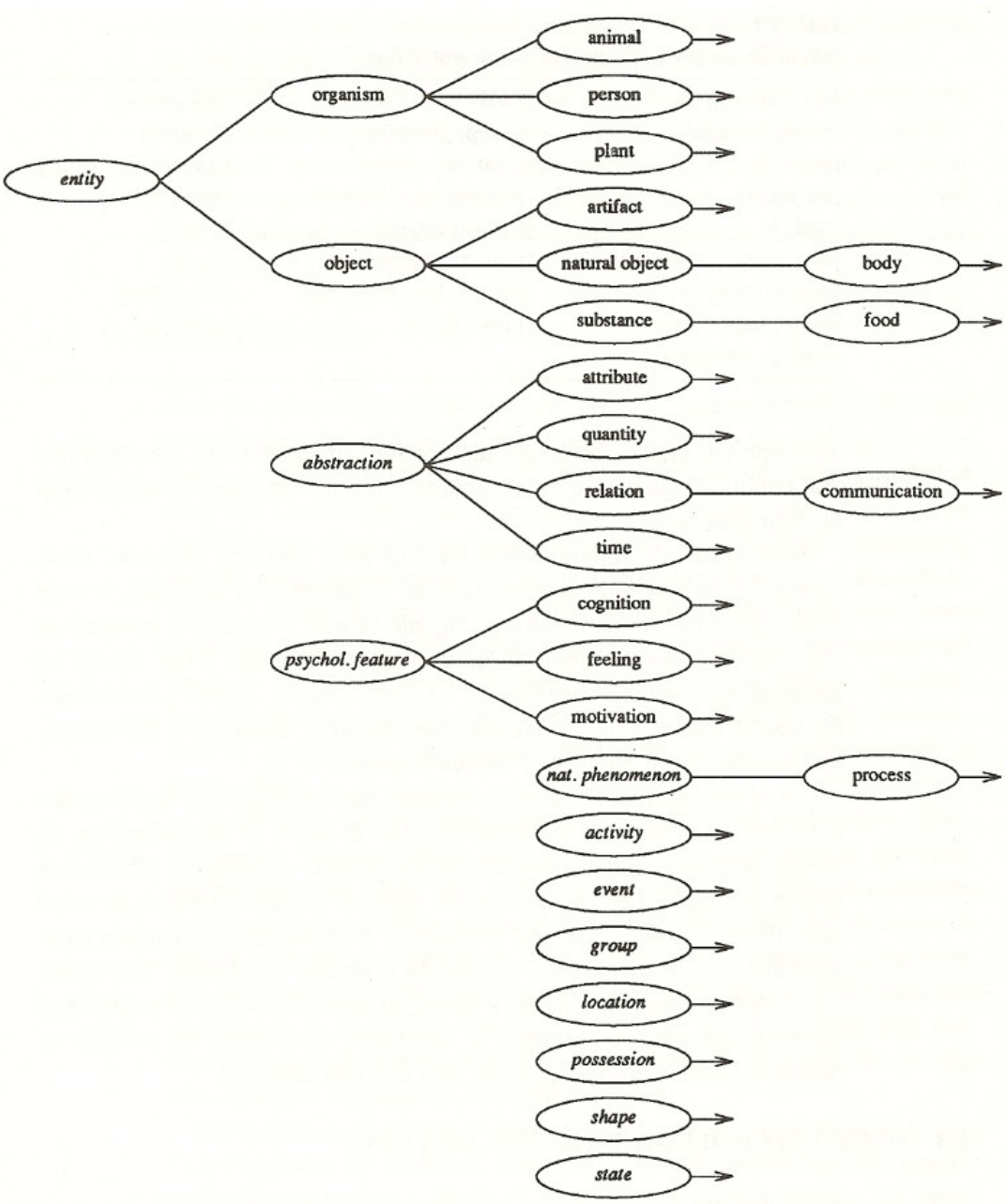

Abbildung 15: WordNet unique beginner (Miller, 1999)

Semantische Verbindungen

Die einzelnen Wörter bzw. der Synsets sind über semantische Relationen miteinander verbunden, wobei es in Abhängigkeit von der Wortart unterschiedliche Verbindungen gibt. Für Substantive werden die Relationen Synonymie, Hyperonymie, Antonymie und Meronymie verwendet. Synonyme sind Wörter, die in einem bestimmten Kontext die gleiche Bedeutung haben (z.B.: Aufzug und Lift).

Hyperonymie ist jene Verbindung, bei dem ein Wort dem anderen übergeordnet ist. Für das Wort Katze ist beispielsweise das Wort Tier ein Hyperonym. Die umgekehrte Relation wird als Hyponym bezeichnet. Antonymie bezeichnet eine gegenteilige Bedeutung, wie z. B.: die Wörter Sieg und Niederlage. Meronymie bezeichnet die Teil-von-Beziehung, z.B.: eine Blüte ist Teil einer Pflanze. Die zu Meronymie entgegengesetzte Beziehung wird Holonymie bezeichnet (vgl. Miller, 1995).

Für Verben gibt es in WordNet die Relationen Synonymie, Hyperonymie, Troponymie und Implikation. Troponymie bedeutet, dass zwei Wörter sehr ähnliche Aktivitäten bezeichnen, die als nahezu gleich zu sehen sind. Troponyme sind beispielsweise marschieren und gehen oder auch flüstern und sprechen. Eine Implikation ist in diesem Kontext eine Aktivität, die eine andere Aktivität bedingt. So ist die Voraussetzung, dass man irgendwo ankommt, dadurch bedingt, dass man dort hinfährt oder hingeht. Für Adjektive und Adverbien werden prinzipiell Synonymie und Antonymie angewandt (vgl. Miller, 1995).

Mit Hilfe von verschiedenen Interfaces ist es möglich, die oben genannten Beziehungen bzw. Wörter, die solche Beziehungen zueinander aufweisen, aus WordNet zu extrahieren. Dies bringt natürlich erhebliche Vorteile für die semantische Analyse von Texten und daher wird WordNet auch das zentrale Element dieser Analyse im System darstellen.

6.3 Weitere Tools und Frameworks

In diesem Abschnitt werden zusätzlich zu WordNet und GATE weitere, für die Umsetzung der in Abschnitt 5 vorgestellten Konzepte benötigte, Tools und Frameworks kurz erläutert.

6.3.1 Java OpenDocument Converter

Der Java OpenDocument Converter (JOD Converter) dient dazu, die unterschiedlichsten Dateiformte in ein gewünschtes Format zu konvertieren. Mit dem JOD Converter können sowohl OpenOffice Formate als auch Microsoft Office Formate konvertiert werden. Darüber hinaus werden PDF, RTF und HTML unterstützt. Eine PDF Datei kann allerdings nur erzeugt werden, das Einlesen einer PDF Datei ist nicht möglich. Ein wesentlicher Nachteil dieses Tools ist jedoch, dass für die korrekte Funktionsweise eine Installation von OpenOffice nötig ist. Weitere Informationen zum JOD Converter finden sich auf der Homepage der Entwickler (vgl. Art of Solving, 2010).

6.3.2 PDFBox

PDFBox ist eine Java PDF Bibliothek, welche das Arbeiten mit PDF – Dateien in Java ermöglicht. Damit ist es möglich, PDF – Dateien zu erzeugen, diese zu

manipulieren und Daten aus vorhandenen Dateien zu extrahieren (vgl. Apache, 2010). Diese Bibliothek wird im System eingesetzt, um PDF – Dateien der weiteren Verarbeitung zugänglich zu machen.

6.3.3 XtraK4Me

XtraK4Me ist ein Algorithmus zur Extraktion von Schlüsselwörtern und Phrasen zum Zwecke der Generierung von Metadaten. Dieser Algorithmus funktioniert für verschiedene Sprachen wie beispielsweise Deutsch, Englisch oder auch Französisch. Dabei verwendet dieser Algorithmus das GATE Framework für die Tokenisierung, die Satzgrenzenerkennung und für die Wortartenerkennung. Zusätzlich werden die Stoppwörter identifiziert, sowie morphologische Analysen und ein Chunking durchgeführt. Abschließend werden anhand von statistischen und lexikalischen Methoden die Keyphrases aus dem Dokument extrahiert (SmILE, 2010).

6.3.4 HTML Cleaner

Der HTML Cleaner ist ein Open Source HTML Parser für Java. Mit Hilfe dieser Bibliothek ist es möglich, auf die einzelnen HTML Element in einer HTML Datei zuzugreifen und diese zu manipulieren. Der HTML Cleaner ermöglicht es darüber hinaus, die Ausgangs HMTL – Datei zu bereinigen bzw. aus diesem bereinigten Dokument eine XML Datei zu erzeugen (vgl. HTML Cleaner, 2010).

6.3.5 JDOM

JDOM ist ein Java API, welches dazu dient, XML in Java zugänglich zu machen. Mit diesem API ist es möglich, XML Dateien zu lesen, sie zu manipulieren und sie wieder auszugegeben bzw. neue XML Dateien zu erzeugen (vgl. JDOM, 2010).

6.4 Zusammenfassung

In diesem Kapitel werden die einzelnen Tools und Frameworks präsentiert, die in weiterer Folge für die Umsetzung der in Kapitel 5 vorgestellten Konzepte verwendet werden.

GATE wird für die grundlegenden Analysen wie Tokenisierung, Wortartenerkennung, Satzgrenzenerkennung usw. eingesetzt. Mit Hilfe von WordNet werden anschließend die semantischen Analysen durchgeführt. Der JOD Converter dient dazu, die unterschiedlichsten Dateiformate der Eingangsdateien in ein einheitliches HTML Format umzuwandeln. Um auch PDF Dateien verwenden zu können, wird mittels der PDFBox der Text aus diesen Dateien extrahiert und ebenfalls in eine HTML Datei umgewandelt. Der HTML Cleaner dient dazu, die essentiellen Daten aus diesen HMTL Dateien zu extrahieren, um anschließend

mittels des JDOM APIs XML Dateien zu erzeugen, welche alle, für die weitere Verarbeitung, benötigten Daten beinhalten. XtraK4Me wird verwendet, um eine zusätzliche Möglichkeit zu bieten, die Auswahl der Schlüsselwörter zu beeinflussen, indem die von diesem Algorithmus ermittelten Schlüsselwörter und Phrasen in den Prozess der Extraktion mit einbezogen werden.

Im nachfolgenden Kapitel wird die konkrete Umsetzung der in Kapitel 5 vorgestellten Konzepte aufgezeigt. Dabei wird auch auf die Integration der in diesem Kapitel präsentierten Tools und Frameworks eingegangen. Darüber hinaus wird die konkrete Funktionalität der Tools, welche im System Anwendung findet, näher erläutert.

.

7 Concept Extractor

In diesem Kapitel wird der Concept Extractor vorgestellt, der aus einem beliebigen natürlich sprachlichen Dokument Konzepte und Daten extrahiert, die den wesentlichen Inhalt des Textes repräsentieren. Der Concept Extractor ist Teil des Automatic Question Creator, welcher ausgehend von diesen extrahierten Daten automatisch Fragen generiert. In diesem Abschnitt werden nun eine detaillierte Darstellung der Funktionsweise des Systems und die konkrete Implementierung der Konzeptextraktion präsentiert. Dabei wird auch gezeigt, wie die in Kapitel 5 aufgezeigten Konzepte umgesetzt werden und es wird auf die Integration der Tools und Frameworks aus Kapitel 6 in das System näher eingegangen. Des Weiteren wird kurz auf Probleme eingegangen, die während der Implementierungsphase aufgetreten sind. Anschließend wird die Benutzeroberfläche vorgestellt und es werden einige Erweiterungsmöglichkeiten bzw. Verbesserungsmöglichkeiten präsentiert.

7.1 Architektur

In Abbildung 16 wird die grundlegende Architektur des Automatic Question Creator gezeigt, wobei nicht grau hinterlegten Module den Concept Extractor darstellen. Das graphische Benutzerinterface dient dabei als Schnittstelle zwischen Benutzer und Applikation. Der Benutzer hat die Möglichkeit, dem System ein Textdokument zuzuführen, welches anschließend die Module Vorverarbeitung, Textanalyse, Datenermittlung und Konzeptextraktion durchläuft.

In den Modulen Vorverarbeitung und Textanalyse werden die externen Tools und Frameworks aus Kapitel 6 verwendet sowie zusätzlich die Erkenntnisse aus dem theoretischen Teil der Arbeit mit eingebracht. Dabei wurden verschiedene, aus diesen Erkenntnissen ableitbare Algorithmen implementiert und in das System integriert. Dieser gesamte Prozess liefert dann als Ergebnis jene Wörter und Phrasen des Textes, welche die wesentlichen Inhalte und Konzepte des Textes repräsentieren sollen. Mittels dieser extrahierten Konzepte werden dann die Fragen generiert und dem Benutzer angezeigt.

Die Implementierung der Fragengenerierung ist nicht Teil dieser Arbeit und daher wird an dieser Stelle auf Lankmayr (2010) verwiesen.

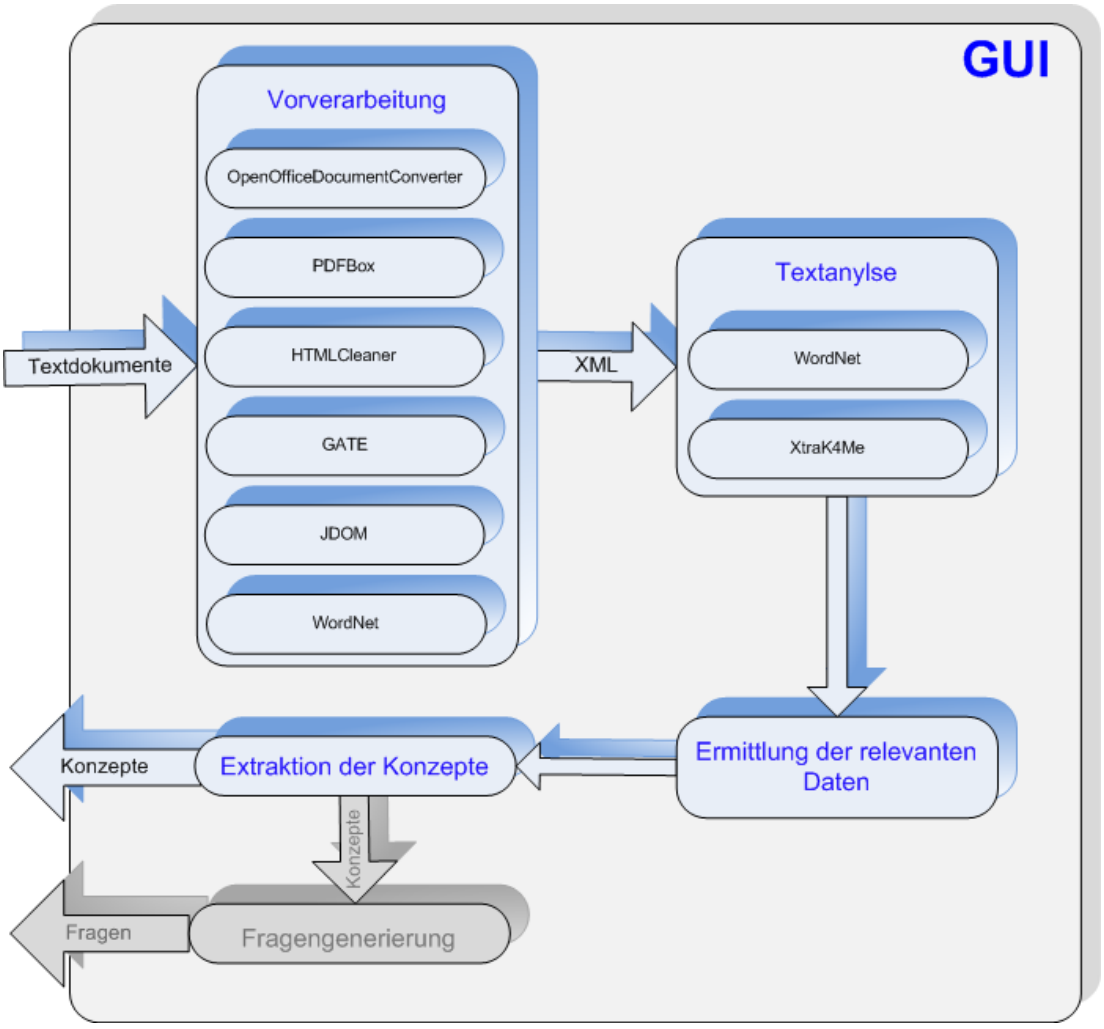

Abbildung 16: Prinzipielle Architektur des Automatic Question Creator

7.2 Implementierung

In diesem Abschnitt werden die konkrete Implementierung der Konzepte sowie die Verwendung der externen Tools und Frameworks näher erläutert. Dieser Abschnitt gliedert sich dabei anhand der Module aus Abbildung 16.

7.2.1 Vorverarbeitung

Die Vorverarbeitung gliedert sich prinzipiell in zwei Teile. Als erstes findet eine Formatkonvertierung statt um anschließend grundlegende morphologische und syntaktische Analysen des Textes durchführen zu können.

7.2.1.1 Formatkonvertierung

Bei der Formatkonvertierung wird das Eingangsdokument in ein geeignetes Format transformiert. Dieses transformierte Format ist in diesem Falle eine XML Datei, die

alle wesentlichen Inhalte und strukturellen Eigenschaften des Ausgangsdokumentes beinhaltet. In Abbildung 17 wird die prinzipielle Architektur der Formatkonvertierung vorgestellt.

Das System unterstützt als Eingangsformate prinzipiell die Dokumenttypen .doc, .odt, .pdf und .html. Darüber hinaus bietet das System die Möglichkeit, eine URL anzugeben. Des Weiteren wäre es möglich, alle Dateitypen, welche vom OpenDocument Converter unterstützt werden und für ein derartiges System sinnvoll sind, in das System zu integrieren. Für die Eingangsformate doc, odt und html ist der Prozess der Formatkonvertierung immer der Selbe. Als erstes wird dabei mittels des OpenDocument Converter die Datei eingelesen und in eine HTML Datei umgewandelt. Die entstehende Datei wird anschließend in ein geeignetes XML Format übergeführt. Dabei wird die Datei mittels des HTML Cleaners bereinigt, wobei spezielle ungeeignete Strukturen aus dieser entfernt werden. Danach werden mit diesem Tool die einzelnen HTML Tags bzw. deren Inhalte aus dem Dokument ausgelesen.

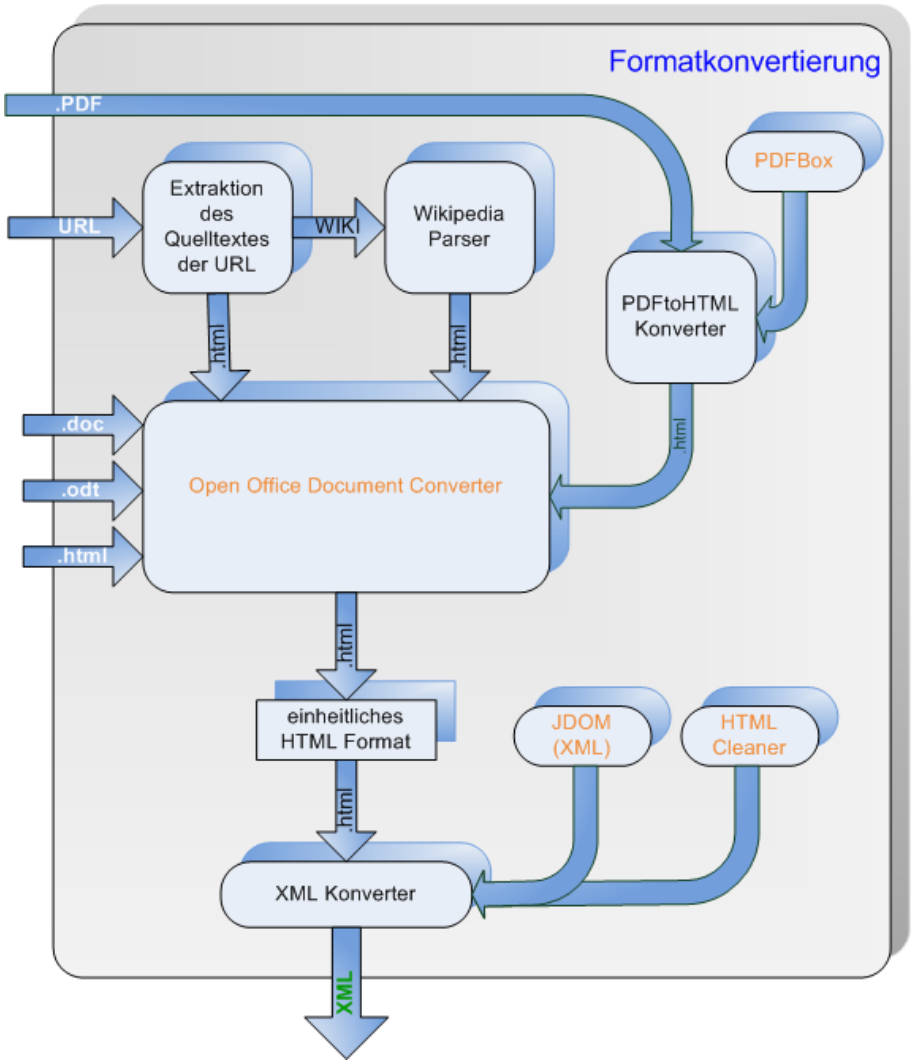

Abbildung 17: Schema der Formatkonvertierung

Diese für die weitere Bearbeitung benötigten Informationen werden somit anschließend mittels des JDOM APIs in ein XML Schema übergeführt und für die weitere Verarbeitung gespeichert. Diese Informationen repräsentieren einerseits die Struktur des Ursprungstextes und andererseits den Text selbst. Als Struktur des Ursprungstextes sind sowohl der Titel des Dokuments, die Gliederung des Textes durch die Überschriften als auch die speziellen Formatierungen und die speziellen Annotationen zu verstehen. Alle weiteren Informationen im Dokument werden verworfen. So werden beispielsweise das eventuell vorhandene Inhaltsverzeichnis, die Literaturliste, Tabellen, Bilder, Bildunterschriften usw. ignoriert. Die beim Prozess der Formatkonvertierung erzeugte XML Datei enthält somit alle für die weitere Verarbeitung essentiell benötigten Informationen.

Falls dem System eine URL angegeben wird, wird versucht, den Quelltext jener Internetseite zu extrahieren, welche diese URL referenziert. Wenn dies gelingt ist die weitere Vorgehensweise ident zu der oben beschriebenen Vorgehensweise für .doc, .odt und .html Dateien. Allerdings gibt es eine Ausnahme. Sollte die (gültige) URL eine Referenz auf eine WIKIPEDIA Seite sein, so wird der extrahierte Quelltext dem sogenannte Wikipedia Parser zur Verfügung gestellt. Dieser Parser entfernt unnötige Informationen aus dem Quelltext, welche sonst nicht herausgefiltert werden und eventuell die Performance des Systems stark beeinflussen könnte. Die mit diesem Parser erzeugte HTML Datei wird wiederum dem OpenDocument Converter zur Verfügung gestellt und der übliche Ablauf der Formatkonvertierung wird fortgesetzt.

Der Wikipedia Parser wurde implementiert, da diese Online Enzyklopädie sehr umfangreich ist und für viele Personen eine wichtige Informationsquelle darstellt. Zusätzlich wird dieser Parser dadurch sinnvoll, dass die in dieser Enzyklopädie enthaltenen Artikel alle einen ähnlichen Aufbau mit sich bringen und daher die Entfernung der nicht verwendbaren Informationen einfach zu gestalten ist. Dieser Prozess erhöht damit die Effizienz und die Zuverlässigkeit des Systems deutlich.

Ein besonderer Fall des Eingangsformates ist das PDF Format, da selbst der OpenDocument Converter keine PDF Dateien verarbeiten kann. Darüber hinaus wurde im Zuge der Recherchen für diese Arbeit kein zuverlässiges und frei verfügbares Programm aufgefunden, welches die für die weitere Verarbeitung benötigten Informationen aus einer PDF Datei auslesen kann. Aus diesem Grund wurde der PDFtoHTML Converter entwickelt. Dieser liest mit Hilfe der PDFBox sämtliche textuellen Informationen aus der PDF Datei aus und versucht mittels vordefinierten Regeln, die relevanten Information, wie beispielsweise Überschriften, Titel usw. im Text zu identifizieren. Diese Informationen werden dann anschließend dazu genutzt, eine geeignete HTML Datei zu erzeugen, welche dann wiederum dem OpenDocument Converter für die weitere Verarbeitung zur Verfügung gestellt wird. Hierbei sei allerdings erwähnt, dass der PDFtoHTML Converter sehr primitiv und unausgereift ist und daher teilweise wenig zufriedenstellende Ergebnisse liefert.

Dies ist bedingt dadurch, dass das Konvertieren einer PDF Datei eine sehr komplexe Aufgabe darstellt und darüber hinaus nicht mit dem zentralen Thema dieser Arbeit einhergeht.

7.2.1.2 Textvorverarbeitung

Im Anschluss an die Formatkonvertierung wird die Textvorverarbeitung durchgeführt. Sie ist Voraussetzung für die nachfolgende Textanalyse. Die prinzipielle Vorgehensweise bei dieser Vorverarbeitung wird in Abbildung 18 dargestellt. Das zentrale Element bei diesem Prozess ist das GATE Tool, woraus hauptsächlich das ANNIE Plug-In Anwendung findet.

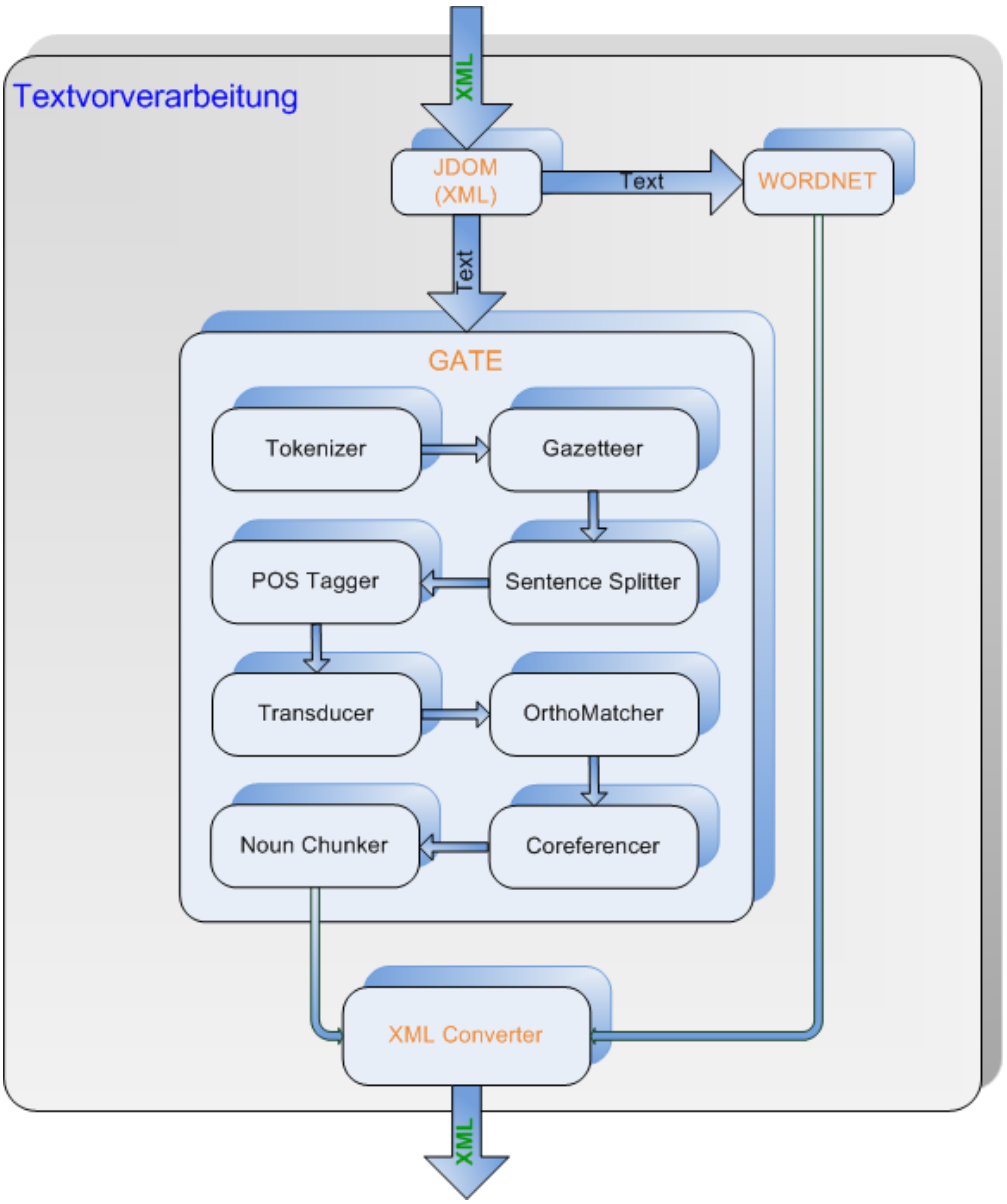

Abbildung 18: Schema der Textvorverarbeitung

Der Input der Textvorverarbeitung ist die bei der Formatkonvertierung erzeugte XML Datei. Aus dieser XML Datei wird dann der Text abschnittsweise extrahiert und dem GATE Tool zur Verfügung gestellt, was bedeutet, dass sowohl Überschriften, als auch die Textkörper selbst jeweils separat mittels GATE analysiert werden. Im Verarbeitungsprozess von GATE selbst werden in diesem Fall die einzelnen Module des ANNIE Plug-Ins auf den Text angewendet. Zusätzlich wird noch der Noun Chunker, welcher einem CEROLE Plug-In entspricht, verwendet. Als erstes wird der Text tokenisiert. Anschließend wird ein sogenannter Gazetteer ausgeführt, der zur Erkennung geographischen Bezeichnungen im Text dient. Danach werden eine Satzgrenzenerkennung und eine Wortartenerkennung durchgeführt. Als nächstes wird der sogenannte Transducer verwendet, der hierbei zur Erkennung von Eigennamen eingesetzt wird. Abschließend werden eine Koreferenzauflösung sowie eine Identifizierung von Nounchunks durchgeführt.

Das Ergebnis der GATE Analyse ist ein spezielles Annotationschema, welches alle gefundenen Informationen zu jedem Wort beinhaltet. Aus diesem Annotationschema werden dann jene Informationen herausgefiltert, die für die weitere Verarbeitung des Textes benötigt werden. Zusätzlich dazu werden für jedes Wort die *unique beginner* aus WordNet ermittelt und diesen Informationen hinzugefügt. Diese Informationen werden dann wiederum in ein geeignetes XML transformiert und in einer Datei abgespeichert.

7.2.2 Textanalyse

Die Textanalyse gliedert sich in drei Abschnitte. Dabei werden strukturelle, statistische und semantische Analysen durchgeführt. Die für die Textanalyse benötigten Daten werden dabei das im Vorverarbeitungschritt erzeugte XML Dokument extrahiert. Für diese Textanalyse wurde eine geeignete interne Datenstruktur eingeführt, die es ermöglicht, für die jeweilige Analysemethode die Einflüsse auf das endgültige Gesamtgewicht zu speichern, um anschließende die Berechnung des Endgewichtes nachvollziehbar und zentral gestalten zu können. Dabei ist zu erwähnen, dass alle nachfolgend beschriebenen Analysemethoden unabhängig voneinander aus dem Prozess der Analyse ausgegliedert werden können. Einzig die statistische Analyse muss durchgeführt werden, da andere Analyseverfahren das dabei ermittelte statistische Gewicht unbedingt benötigen.

7.2.2.1 Strukturelle Analyse

Die strukturelle Analyse befasst sich mit dem Auffinden von speziellen Strukturen in den Texten. Diese speziellen Strukturen sind besondere Überschriften in den Dokumenten, spezielle Formatierungen, Koreferenzen sowie die speziellen Annotierungen.

Zur strukturellen Analyse würde prinzipiell auch die Unterteilung der Dokumente in einzelne Abschnitte gehören. Diese Unterteilung findet allerdings schon in den Vorverarbeitungsschritten statt, wobei der Einfachheit halber ein Abschnittswechsel mit dem Auftreten einer neuen Überschrift der höchsten Kategorie gleichzusetzen ist.

Spezielle Überschriften sind vor allem *Kurzfassung* und *Schlüsselwörter*. Diese werden bei der nachfolgenden statistischen Analyse nicht berücksichtigt, d. h. die Auftrittshäufigkeit eines Wortes in der Kurzfassung oder in den Schlüsselwörtern wirkt sich nicht auf das statistische Gewicht aus bzw. die Wörter besitzen in diesen Abschnitten kein statistisches Gewicht. Allerdings haben diese Wörter sehr wohl einen Einfluss auf das Endgewicht. So bekommt jedes Wort im Text, dass auch in der Kurzfassung oder in den Schlüsselwörtern vorkommt, einen bestimmten Faktor zugewiesen, der sich positiv auf das Endgewicht auswirkt (siehe Abschnitt 7.2.2.3).

Spezielle Formatierungen von Wörtern, wie beispielsweise fett oder kursiv formatierte bzw. unterstrichene Wörter werden ebenfalls bei der strukturellen Analyse berücksichtigt. Dies beruht auf die Annahme, dass Wörter, die solch eine Formatierung aufweisen mit relativ großer Wahrscheinlichkeit wichtig sind. Diesen Wörtern wird ebenfalls ein bestimmter positiver Wert ($w_{formatting}$) als Formatierungsgewicht zugewiesen.

Darüber hinaus werden bei der strukturellen Analyse auch die speziellen Annotationen, welche von GATE identifiziert werden, mit einbezogen. Dabei erhält jedes Wort, das Teil einer speziellen Annotation ist, einen vordefinierten und einstellbaren Faktor ($w_{annotation}$) als Annotationsgewicht zugewiesen. Diese Faktoren können für jede spezielle Annotation individuell bestimmt werden und wirken sich folglich auf alle Wörter aus, die als spezielle Annotation identifiziert werden. So werden beispielsweise Wörter, die zur speziellen Annotation *Person* gehören mit dem Faktor α gewichtet, Wörter die zur Kategorie Datum gehören werden mit dem Faktor β gewichtet, wobei die Faktoren α und β frei wählbar sind.

7.2.2.2 Statistische Analyse

Als statistische Analyse des Textes wird hierbei die Ermittlung der Häufigkeit des Auftretens eines Wortes im Text bezeichnet. Dabei ist zu beachten, dass einerseits die Häufigkeit standardmäßig nur abschnittsweise ermittelt wird und andererseits die Ermittlung der Häufigkeit eine Grundformreduktion der Wörter bedingt. Diese Grundformreduktion wird bereits im Vorverarbeitungsschritt mittels eines Stemming Algorithmus (Porter Stemmer) durchgeführt.

Bei der Ermittlung wird also jedes Vorkommen eines auf die Grundform reduzierten Wortes ermittelt. Dabei werden ausschließlich nur jene Wörter gezählt, die von GATE als Substantive und Verben erkannt werden. Als Häufigkeitsmaß wird

dabei eine spezielle Version der inversen Dokumentfrequenz (IDF) benutzt. Dabei wird im Concept Extractor die Anzahl von jedem Wort in einem Abschnitt mit der Anzahl des am häufigsten im Abschnitt vorkommenden Wortes normiert. Dies hat zur Folge, dass dem häufigsten im Abschnitt vorkommenden Wort das statistische Gewicht 1 zugewiesen wird. Der Hauptzweck dieser Normierung ist, den Einfluss der Länge eines Dokuments bzw. eines Abschnittes auf die Ermittlung des statistischen Gewichtes und damit auf die Berechnung des Endgewichtes zu reduzieren.

Des Weiteren wird bei dieser Analyse bereits vor der Normierung der Einfluss der Koreferenzen auf das statistische Gewicht berücksichtigt. Dabei wird das statistische Gewicht des Wortes, auf welches sich die Koreferenz bezieht um eins erhöht, da durch diese Koreferenz dieses Wort prinzipiell einmal öfters vorkommt. Dieselbe Prozedur bietet sich auch für Akronyme und Anaphorische Ausdrücke an. Dies ist allerdings in diesem System nicht berücksichtigt worden.

7.2.2.3 Semantische Analyse

Der erste Schritt bei der semantischen Analyse ist die Ermittlung der ähnlichen Wörter für jedes Wort. Dies wird mit Hilfe von WordNet durchgeführt, wobei zusätzlich auch ein Ähnlichkeitsmaß für diese Wortpaare bestimmt wird. Dabei stehen beim Concept Extractor zwei verschieden Algorithmen zur Auswahl, welche frei wählbar sind leicht unterschiedliche Ergebnisse liefern.

Unabhängig vom gewählten Ähnlichkeitsalgorithmus werden dabei für jedes Substantive alle ähnlichen Substantive im Dokument abschnittsweise ermittelt und geeignet gespeichert. Ein Wort ist einem anderen Wort ähnlich, wenn das Ähnlichkeitsmaß der Wörter über einem vordefinierten Schwellenwert liegt. Dieser Schwellwert kann vor der Ähnlichkeitsberechnung bestimmt werden. Das hierbei verwendete Ähnlichkeitsmaß wird von den oben erwähnten Algorithmen jeweils für ein bestimmtes Wortpaar zurückgeliefert.

Eine zusätzliche Bedingung für die Ähnlichkeit von zwei Wörtern ist, dass sich die beiden Wörter mindestens einen unique beginner teilen müssen. Dies beruht darauf, dass ein Wort, nur dann zu einem anderen ähnlich sein kann, wenn sie sich die gleiche WordNet Kategorie teilen. Diese Ähnlichkeiten beeinflussen das endgültige Gesamtgewicht auf unterschiedlichste Art und Weise.

Diese Einflüsse werden nachfolgend kurz beschrieben:

1. *Wortähnlichkeit*: Als erstes wird ein Wortähnlichkeitsgewichtsfaktor gebildet, der das statistische Gewicht der ähnlichen Wörter eines Wortes auf das Wort selbst berücksichtigt. Dabei wird die zuvor ermittelte Ähnlichkeit ebenfalls in

die Berechnung mit einbezogen. Der Wortähnlichkeitsgewichtsfaktor wird nach Formel 7.1 ermittelt.

$$w_{sim}(i) = \sum_{j=1}^{n} w_{stat}(j) * sim_{(i,j)} \quad 7.1$$

Dabei bezeichnet $w_{sim}(i)$ den Wortähnlichkeitsgewichtsfaktor für das Wort i. n ist die Anzahl der ähnlichen Wörter von Wort i, $w_{stat}(j)$ ist das statistische Gewicht des j - ten ähnlichen Wortes von Wort i und $sim_{(i,j)}$ ist gleich dem Ähnlichkeitsmaß zwischen Wort i und dem j - ten ähnlichen Wort von Wort i.

2. *Ähnlichkeit mit dem Titel*: Hierbei wird der Titelgewichtsfaktor berechnet. Dieser Faktor soll die Ähnlichkeit von jedem Wort zum Titel ausdrücken. Dabei wird jedem Wort jener Wert zugewiesen, der sich nach Formel 7.2 berechnet.

$$w_{title}(i) = \sum_{j=1}^{n} factor_{title} * sim_{(i,j)} \quad 7.2$$

Dabei ist $w_{title}(i)$ der Titelgewichtsfaktor für das Wort i. n ist gleich der Anzahl der Wörter im Titel, welche zu Wort i ähnlich sind. $sim_{(i,j)}$ ist gleich die Ähnlichkeit des Wortes i mit dem j - ten ähnlichen Wort im Titel. Der Faktor $factor_{title}$ ist ein frei einstellbarer Parameter, der die Größe der Auswirkungen des Titelgewichtsfaktors im Endgewicht charakterisieren soll.

3. *Ähnlichkeit mit Überschriften*: Hierbei wird der Überschriftengewichtsfaktor für jedes Wort ermittelt. Dabei wird analog zu der Berechnung der Ähnlichkeit mit dem Titel die Ähnlichkeit eines Wortes zur Überschrift, in dessen Textkörper sich das das Wort befindet, anhand von Formel 7.3 berechnet.

$$w_{headline}(i) = \sum_{j=1}^{n} factor_{headline} * sim_{(i,j)} \quad 7.3$$

Dabei ist $w_{headline}(i)$ der Überschriftengewichtsfaktor für das Wort i. n ist gleich der Anzahl der Wörter in der Überschrift, welche zu Wort i ähnlich sind. $sim_{(i,j)}$ ist in diesem Fall gleich der Ähnlichkeit des Wort i mit dem j - ten ähnlichen Wort in der Überschrift. Der Faktor $factor_{headline}$ ist ein frei einstellbarer Parameter, der die Auswirkungen des Überschriftengewichtsfaktors im Endgewicht bestimmt.

4. *Ähnlichkeit mit Kurzfassung und Schlüsselwörtern*: Dabei werden die Faktoren berechnet, die die Ähnlichkeit eines Wortes zu den Wörtern in Kurzfassung und Schlüsselwörter repräsentieren. Die Berechnungen erfolgen nach den Formeln 7.4 und 7.5.

$$w_{abstract}(i) = \sum_{j=1}^{n} factor_{abstract} * sim_{(i,j)} \quad 7.4$$

$$w_{keywords}(i) = \sum_{j=1}^{n} factor_{keywords} * sim_{(i,j)} \quad 7.5$$

Dabei ist $w_{abstract}(i)$ der Kurzfassungsgewichtsfaktor und $w_{keywords}(i)$ der Schlüsselwörtergewichtsfaktor. n ist die Anzahl der zum Wort i ähnlichen Wörter in der Kurzfassung bzw. in den Schlüsselwörtern. Sollte sich das Wort selbst in diesen Abschnitten befinden, so ist der Ähnlichkeitsfaktor $sim_{(i,j)}$ gleich eins, was bedeutet, dass die Faktoren $factor_{abstract}$ bzw. $factor_{keywords}$ vollständig in die Gewichtsfaktoren dieses Wortes mit einbezogen werden. $sim_{(i,j)}$ repräsentiert das Ähnlichkeitsmaß zwischen Wort i und Wort j.

5. *Ähnlichkeit Überschriften mit Titel*: Dabei wird die Ähnlichkeit einer Überschrift mit dem Titel berücksichtigt. Das bedeutet, dass einem Wort, welches im Textkörper einer Überschrift auftritt und diese Überschrift Wörter beinhaltet, welche zusätzlich im Titel vorkommen bzw. in den ähnlichen Wörtern der Titelwörter auftreten, ein höherer Überschriftengewichtsfaktor zugewiesen wird. Dies wird durch die Adaption des Faktors $factor_{headline}$ in Formel 7.3 (Berechnung der Ähnlichkeit eines Wortes mit einer Überschrift) bewerkstelligt. Dabei wird dieser Faktor nach Formel 7.6 abgeändert.

$$factor_{headline,new} = factor_{headline,old} * (factor_{title,headline} * 2) \quad 7.6$$

$$factor_{title,headline} = \frac{\sum_{i=1}^{n} \sum_{j=m}^{m} sim(word_h(i), word_t(j))}{n * m} \quad 7.7$$

Dabei ist $factor_{headline,old}$ gleich dem $factor_{headline}$ aus Formel 7.3. $word_h(i)$ ist das i – te Substantive der Überschrift und $word_t(j))$ ist das j – te Substantive des Titels. n ist gleich der Anzahl der Hauptwörter in der Überschrift und m ist die Anzahl der Substantive im Titel. Der Ausdruck $sim(word_h(i), word_t(j))$ entspricht der mittels WordNet ermittelten Ähnlichkeit von Wort i aus der Überschrift mit dem Wort j aus dem Titel. Die Multiplikation mit dem Faktor zwei begründet sich damit, dass der Faktor

$factor_{title,headline}$ eine Zahl zwischen null und eins ist. Dass bedeutet, dass im besten Fall, d. h. die Wörter im Titel sind exakt die Wörter in der Überschrift, der $factor_{headline}$ für die Berechnung des Überschriftengewichtsfaktors verdoppelt wird, da der ermittelte Faktor $factor_{title,headline}$ in diesem Fall gleich eins ist.

6. *Rekursive Wortähnlichkeit*: Dabei werden die einzelnen Wörter nicht nur auf die Ähnlichkeit zu anderen überprüft, sondern es werden auch die ähnlichen Wörter der ähnlichen Wörter in die Berechnung dieses rekursiven Gewichtsfaktors mit einbezogen. Die Berechnung dieses Faktors erfolgt nach Formel 7.8.

$$w_{recursive}(i) = \sum_{j=1}^{n} (w_{stat}(j) * sim_{(i,j)} * \sum_{k=1}^{m} w_{stat}(k) * sim_{(j,k)}) \quad 7.8$$

Der Ausdruck $w_{recursive}(i) = \sum_{j=1}^{n} w_{stat}(j) * sim_{(i,j)}$ ist analog zu Formel 7.1. Bei der Berechnung des rekursiven Gewichtsfaktors werden aber zusätzlich für jedes zu Wort w_i ähnliche Wort w_j die ähnlichen Wörter ermittelt und in die Berechnung mit einbezogen. Dabei ist m die Anzahl der ähnlichen Worte von Wort w_j und $sim_{(j,k)}$ die Ähnlichkeit von Wort w_j mit Wort w_k.

7. *Kategoriegewicht*: Für die Bestimmung des Kategoriegwichtes ($w_{category}$) werden die zuvor ermittelten unique beginner von jedem Wort verwendet. Die Berechnung dieses Gewichtes erfolgt mittels Formel 7.9.

$$w_{category}(i) = \frac{1}{n} * \sum_{j=1}^{n} factor_{beginner}(j) \quad 7.9$$

Dabei ist n die Anzahl der unique beginner von Wort i, und der Faktor $factor_{beginner}(j)$ ist ein vordefinierbarer und frei wählbarer Wert für jeden unique beginner.

8. XtraK4Me *Keyphrases*: Dabei wird der in Abschnitt 6.3.3 vorgestellt Algorithmus benutzt, um aus einem Dokument relevante Daten zu extrahieren. Diesen dabei extrahierten Wörter und Phrasen wird dabei vom Algorithmus ein Maß für die Relevanz zugeordnet, welches als $w_{keyphrase}$ in das System integriert wird.

7.2.3 Ermittlung relevanter Daten

Bei der Ermittlung der relevanten Daten werden die einzelnen in den vorhergehenden Schritten berechneten Werte nach Formel 7.10 zusammengefügt, um das Gesamtgewicht eines Wortes zu erhalten.

$$w\,(i) = w_{stat} * \big(1 + \; w_{sim}(i) + \; w_{title}(i) + w_{headline}(i) + w_{abstract}(i) + w_{keywords}(i) + \; w_{annotation}(i) + \; w_{category}\,(i) + \; w_{formatting}(i) + \; w_{keyphrase}(i)\,\big) + w_{recursive}(i) \qquad 7.10$$

Alle Wörter, die abschließend ein Gesamtgewicht haben, dass kleiner ist als ein vordefinierter Schwellwert werden verworfen. Alle übrigen Wörter werden nach anschließend dem Gesamtgewicht nach sortiert und der Median gebildet. Alle Wörter, die ein Gewicht haben, welches größer ist als der Median, werden bei der nachfolgenden Extraktion der Konzepte berücksichtigt.

7.2.4 Extraktion der Konzepte

Dabei wird für jedes der im vorangegangen Schritt ermittelten Worte in einem Absatz das am höchsten gewichtete Auftreten dieses Wortes gesucht, da die Möglichkeit besteht, dass verschiedenen Vorkommen von ein und demselben Wort in einem Absatz unterschiedliche Gewichte haben können. Dies kann durch fehlerhaftes Erkennen von Koreferenzen durch Gate oder auch durch die speziellen Formatierungen auftreten, welche sich immer nur auf das Wort selbst beziehen. Nach der Ermittlung der am höchsten gewichteten Wörter werden für diese Wörter passende Phrasen (Nounchunks) gesucht, da diese im Allgemeinen eine höhere Aussagekraft haben als die einzelnen Wörter. Diese ermittelten Phrasen werden dann abschließend dem Benutzer angezeigt.

7.2.5 Probleme bei der Implementierung

Die bei der Implementierung der verschiedenen in diesem Abschnitt vorgestellten Methoden und Algorithmen entdeckten Probleme werden in diesem Abschnitt kurz beschrieben.

Durch das Fehlen eines universellen Formatkonverters kommt der OpenDocument Converter zum Einsatz. Dieser unterstützt viele Dateiformaten hat aber auch zwei wesentliche Nachteile. Einerseits kann dieser keine PDF Dateien einlesen, was zur Implementierung des PDFtoHTML Konverters führte, andererseits bedingt der Konverter eine vollständige Installation von OpenOffice. Darüber hinaus ist der PDFtoHTML Konverter wie bereits erwähnt relativ primitiv und liefert teilweise relativ schlechte Ergebnisse.

Des Weiteren haben viele der verwendeten Tools und Frameworks große Probleme mit Texten, die ungewöhnliche Abfolgen von Symbolen, Zeichen bzw. Sonderzeichen beinhalten. Daher wird im Concept Extractor bereits sehr früh versucht, diese Strukturen und Sonderzeichen aus den Eingangstexten herauszufiltern. Allerdings ist Anzunehmen, dass es noch viele weitere solcher Strukturen und spezielle Abfolgen existieren, welche Probleme verursachen und während der Testphase nicht erkannt wurden. Diese konnten somit in diesem Fall nicht behandelt werden.

Zusätzlich traten bei der GATE Annotierung auch Probleme mit falsch erkannten Annotierungen auf. Diese falschen Daten beeinflussen den ganzen Extraktionsprozess in einer negativen Art und Weise und verursachen dadurch oftmals falsche bzw. ungenügende Ergebnisse. Des Weiteren wurde auch das Problem erkannt, dass durch dieses Annotierungen bzw. die gesonderte Behandlung der Annotierungen ebenfalls oftmals falsche Ergebnisse erzielt wurden.

Weitere Probleme traten bei der Integration der einzelnen Tools und Frameworks in das System auf. Es ist oftmals relativ schwierig, diese zu integrieren, oftmals bedingt durch schlechte oder nicht vorhandene Dokumentationen. Des Weiteren benötigen diese Programme des Öfteren spezielle Bibliotheken bzw. oftmals die gleichen Bibliotheken in unterschiedlichen Versionen. Dies führte dann oft zu Konflikten, die nur sehr schwer aufzulösen sind.

Auch bei der Verwendung von WordNet traten Probleme auf. Dabei wurde der Extraktionsprozess vor allem durch das Fehlen von Wörtern in der Datenbank von WordNet behindert. Dies gilt vor allem für nicht alltäglich verwendete Wörter des englischen Sprachgebrauchs und insbesondere für die Inkonsistenz der Integration der Eigennamen. Es sind zwar viele Eigennamen in WordNet integriert und die Verbindungen dieser mit anderen Einträgen funktioniert bestens, allerdings kann diese Datenbank klarerweise nicht alle Namen beinhalten. Ungeeignete Ergebnisse sind dann oftmals auf dieser Tatsche begründet.

Des Weiteren ist Anzumerken, dass viele der implementierten Algorithmen sehr rechenintensiv sind. Dies hat zur Folge, dass die Analyse eines Dokuments in Abhängigkeit von dessen Länge einige Minuten dauern kann.

7.3 Sichtweise des Benutzers

Nach dem Starten des Programms wird dem Benutzer der Einstiegsbildschirm angezeigt (vgl. Abbildung 19). Dieser Bildschirm gliedert sich in ein Navigationsfenster auf der linken Seite und in ein Anzeigefenster. Zusätzlich gibt es eine Menüleiste, welche die grundlegenden Operationen wie neue Datei einfügen, laden, speichern usw. beinhaltet.

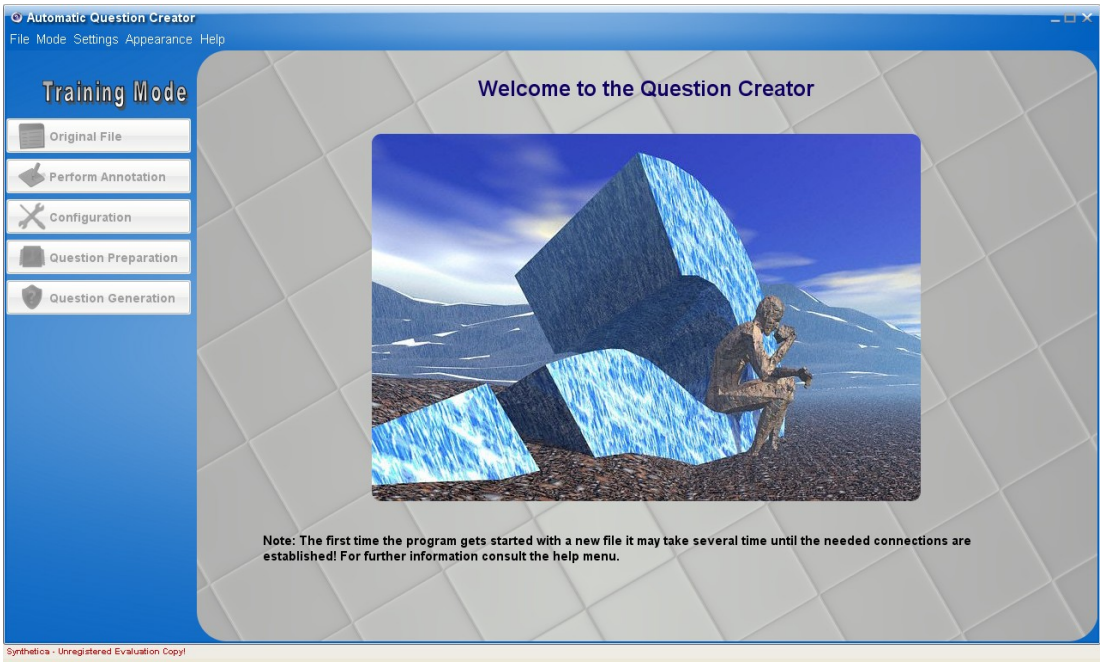

Abbildung 19: Automatic Question Creator: GUI

Nach dem laden einer neuen Datei bietet sich dem Benutzer die Möglichkeit, die Annotierung des Textes durchführen zu lassen. Die Ergebnisse dieser Annotation zeigt Abbildung 20. Diese werden als Text inklusive den dazugehörenden Annotationen im Anzeigefenster dargestellt. Dabei werden die die einzelnen Annotationstypen mittels unterschiedlicher Farben verdeutlicht. Zusätzlich bietet sich dem Benutzer die Möglichkeit, sich den Text auch in der gestemmten Form sowie in Form der verwendeten XML Repräsentation anzeigen zu lassen.

Abbildung 20: Beispiel eines annotierten Textes

Im nächsten Schritt wird das System für die Textanalyse und die damit einhergehende Gewichtsberechnung konfiguriert. Dabei bieten sich dem Benutzer zwei unterschiedliche Möglichkeiten, diese Konfiguration zu gestalten. Die erste Möglichkeit zeigt Abbildung 21.

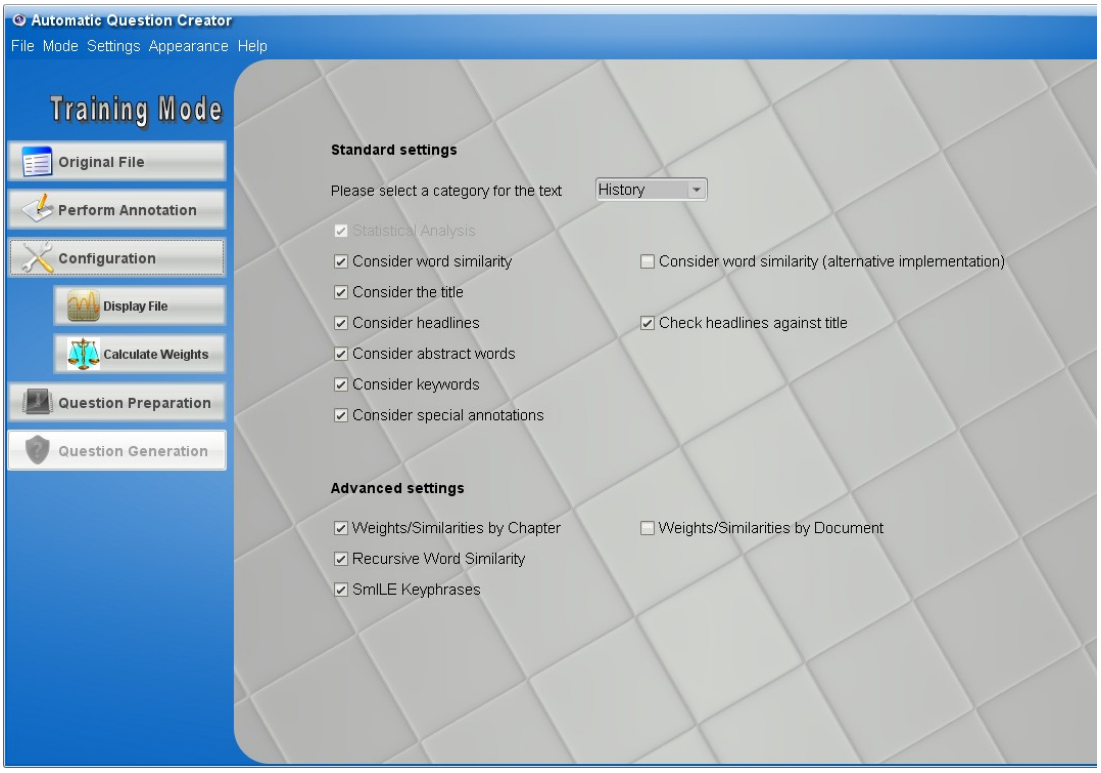

Abbildung 21: Algorithmen Auswahl

In diesem Auswahlfenster können die Benutzer jene Algorithmen wählen, die für die Analyse verwendet werden sollen. Dabei ist zu beachten, dass die statistische Analyse immer durchgeführt werden muss, da diese die Voraussetzung für die weiteren Analyseverfahren darstellt. Die Benutzer können dabei auch den Ähnlichkeitsalgorithmus bestimmen, sowie angeben, ob die Berechnungen kapitelweise durchgeführt werden sollen oder ob das gesamte Dokument in einem Schritt analysiert wird. Hierbei ist zu erwähnen, dass mit der Kapitelmethode im Normallfall bessere Ergebnisse erzielt werden können.

Der Benutzer hat bei diesen Einstellungsoptionen also die Möglichkeit, die Integration von Titel, Kurzfassung, Schlüsselwörter, und Überschriften individuell zu gestalten, um so gegebenenfalls den Prozess der Konzeptextraktion auf die Art und Struktur des Dokuments anzupassen.

Des Weiteren hat der Benutzer die Möglichkeit, eine bestimmte Kategorie vorzugeben. Die Kategorie beeinflusst dabei die Faktoren, die bei der Gewichtsberechnung mit einbezogen werden. Diese Faktoren bzw. deren Manipulation ist gleichzeitig die zweite Option zur Konfiguration des Systems. Diese Möglichkeit wird in Abbildung 22 dargestellt.

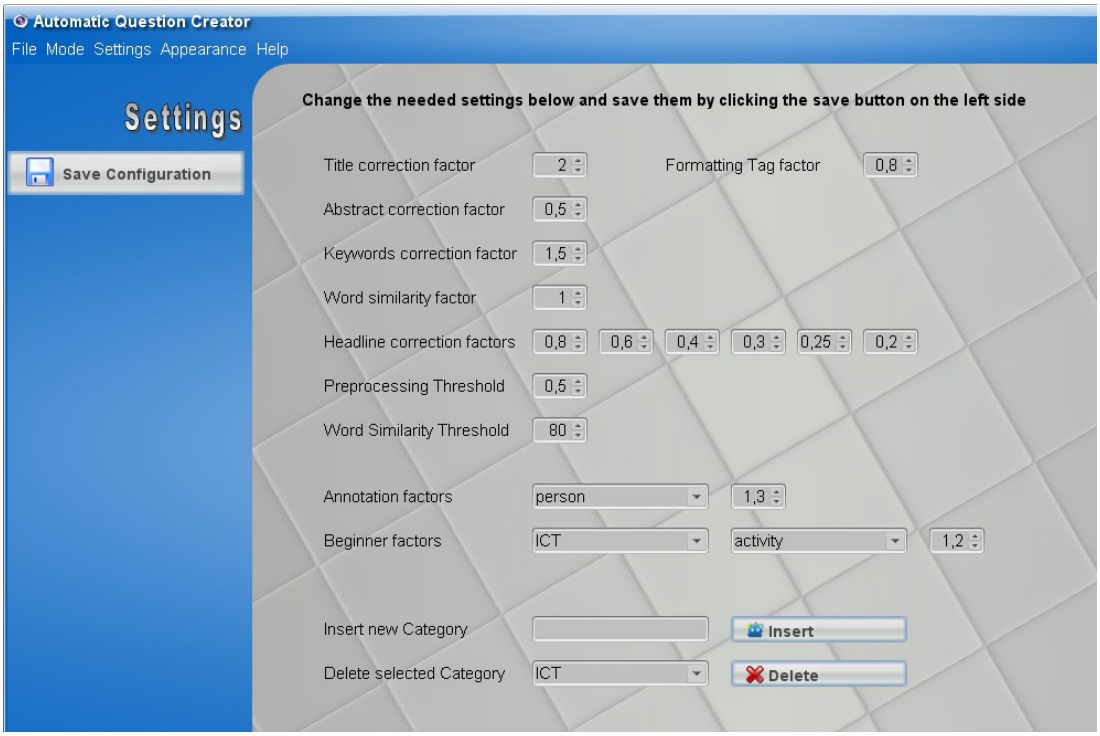

Abbildung 22: Konfiguration der einzelnen Gewichtsfaktoren

In diesem Konfigurationsmenü können die einzelnen Faktoren, wie beispielsweise der Titelfaktor, die Überschriftenfaktoren usw. verändert werden. Diese Änderungen wirken sich dann bei der nachfolgenden Gewichtsberechnung

auf das Endgewicht aus. Des Weiteren bietet das System hierbei die Möglichkeit, den Einfluss der speziellen Annotationen sowie der unique beginner gezielt zu verändern. Dies dient dazu, auf eventuell spezielle Bedürfnisse, die ein Text einer speziellen Wissenskategorie mit sich bringt, in die Berechnung der Gewicht mit einfließen lassen zu können.

Darüber hinaus kann der Wert für das Ähnlichkeitsmaß verändert werden, welches bestimmt, wie ähnlich ein Wort zu einem anderen sein muss, um als ähnliches Wort klassifiziert zu werden. Je niedriger dieser Faktor, desto mehr Wörter werden als ähnlich bestimmt, da die semantischen Verbindungen dann weitläufiger sein können und daher mehr Wörter als ähnlich erkannt werden. Der *Preprocessing Threshold* hingegen ist der Wert, der angibt, ob ein Wort abschließend als semantisch relevant klassifiziert wird, falls das Gesamtgewicht über diesem Schwellwert liegt. Je höher dieser Wert, desto weniger Wörter werden dementsprechend extrahiert und dem Nutzer angezeigt. Zusätzlich können die Benutzer eigene Kategorien mit den von ihnen gewünschten Einstellungen erstellen und abspeichern bzw. diese auch wieder löschen.

Nach dem optionalen konfigurieren des Systems können die Nutzer die Textanalyse und die damit einhergehende Gewichtsberechnung starten. Das Ergebnis dieser Berechnung zeigt Abbildung 23. Wie in dieser Abbildung ersichtlich, wird die Farbgebung der speziellen Annotationen auf Grund der Übersichtlichkeit weiter beibehalten.

Abbildung 23: Gewichteter Text

Die Benutzer haben bei der Anzeige des gewichteten Textes nun die Möglichkeit, sich genauere Information über die Gewichtsberechnung anzeigen zu lassen. Dafür muss nur der Cursor über ein bestimmtes Wort platziert werden, wodurch ein Tooltip angezeigt wird, welcher jene Informationen beinhaltet, die für dieses Wort relevant sind. Im Beispiel in Abbildung 23 ist für die Anzeige des Tooltips das Wort *Austria* ausgewählt worden. Als erstes wird die zu diesem Wort gehörende Nominalphrase (*Prussia and Austria*) angezeigt. Anschließend wird das statistische Gewicht gezeigt. Für dieses Wort ist dieses Gewicht 1, da es das am häufigsten vorkommende Wort in diesem Abschnitt ist. Des Weiteren wird die Exakte Anzahl des Wortes in diesem Abschnitt präsentiert, welche in diesem Fall fünfzig beträgt.

Die *Category* zeigt den unique beginner des Wortes *Austria* in WordNet an. In diesem Fall ist dieser eindeutig, was bedeutet, dass dieses Wort nur eine Bedeutung in WordNet aufweist. Dies ist im Allgemeinen nicht der Fall, da ein Wort oftmals mehrere Bedeutungen hat. Danach sind die ähnlichen Wörter und das dazugehörende Ähnlichkeitsmaß angeführt. Das aus diesen Wörtern und Werten berechnete Gewicht für die Wortähnlichkeit wird unter diesen Wörtern angezeigt.

Die danach folgenden Einträge im TooltipText zeigen, dass das Wort im Titel, in der Kurzfassung und in den Schlüsselwörtern vorkommt. Des Weiteren ist erkennbar, dass dieses Wort eine spezielle Annotation (Ort) ist und dass durch diese Kategorie bzw. den dahinterstehenden Faktor das Endgewicht ebenfalls beeinflusst wird. Das rekursive Gewicht ist bei diesem Wort auf Grund der zugrundeliegenden Berechnungsmethode vernachlässigbar. Das *Keyphrase Weight* zeigt an, dass der XtraK4Me Algorithmus das Wort ebenfalls als wichtiges Schlüsselwort ausgewählt hat. Zusätzlich wird die von diesem Algorithmus bestimmte Relevanz für dieses Wort angezeigt. Abschließend wird in diesem Tooltip dann das aus den anderen Faktoren und Werten berechnete Gesamtgewicht für das ausgewählte Wort angezeigt.

Nach der Textanalyse und Gewichtsberechnung haben die Benutzer die Möglichkeit, die *Question Preparation* durchzuführen. Diese dient dazu, die geeigneten Wörter und Phrasen zu extrahieren, die als semantisch relevante Daten identifiziert wurden. Diese Wörter und Phrasen werden, wie in Abbildung 24 ersichtlich für jeden Abschnitt gesondert dargestellt. Dabei wird, wenn der Cursor über einer Phrase platziert wird, wiederum ein Tooltip angezeigt. Dieser beinhaltet neben dem Wort, welches ausgewählt und um welches die Phrase gebildet wurde, auch das Gesamtgewicht dieser Phrase.

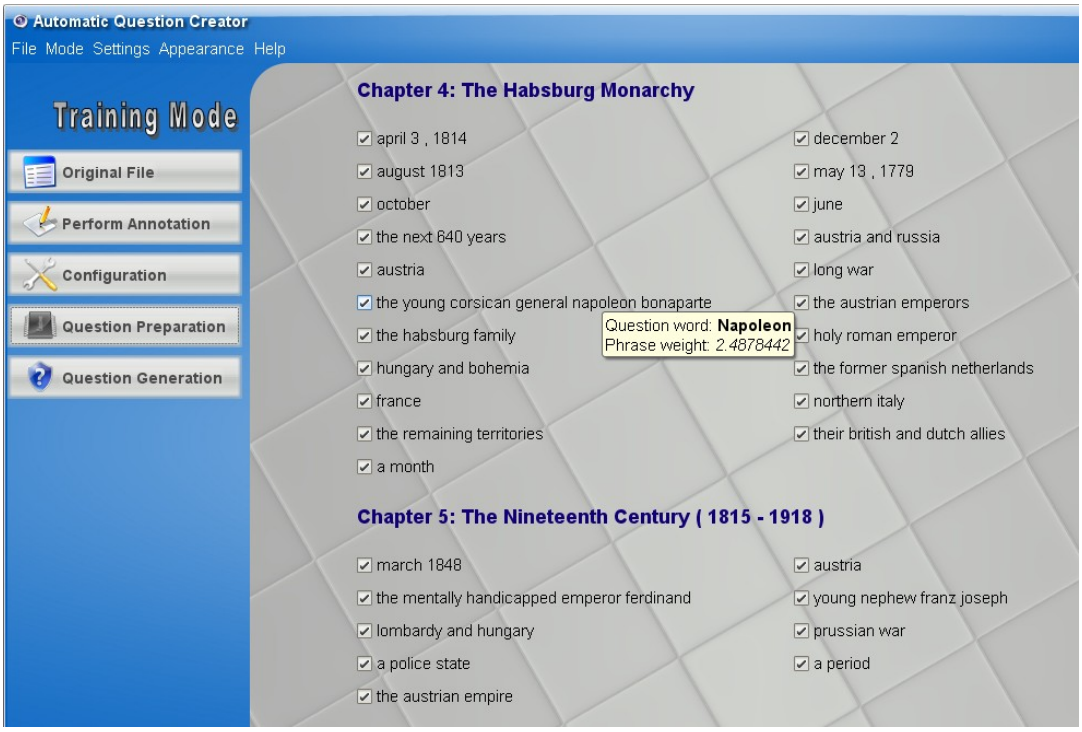

Abbildung 24: Extrahierte Wörter und Phrasen

Abschließend bietet sich dem Benutzer die Möglichkeit, sich in Abhängigkeit dieser Phrasen geeignete Fragen zu den unterstützten Fragetypen erzeugen zu lassen. Abbildung 25 zeigt drei mit dem Automatic Question Creator erzeugte Lückentextfragen. Des Weiteren werden in diesem System die Fragetypen Multiple Choice, Single Choice und offene Fragen unterstützt. Für eine detaillierte Beschreibung des Prozess zur Fragengenerierung sei nochmals auf Lankmayr (2010) verwiesen.

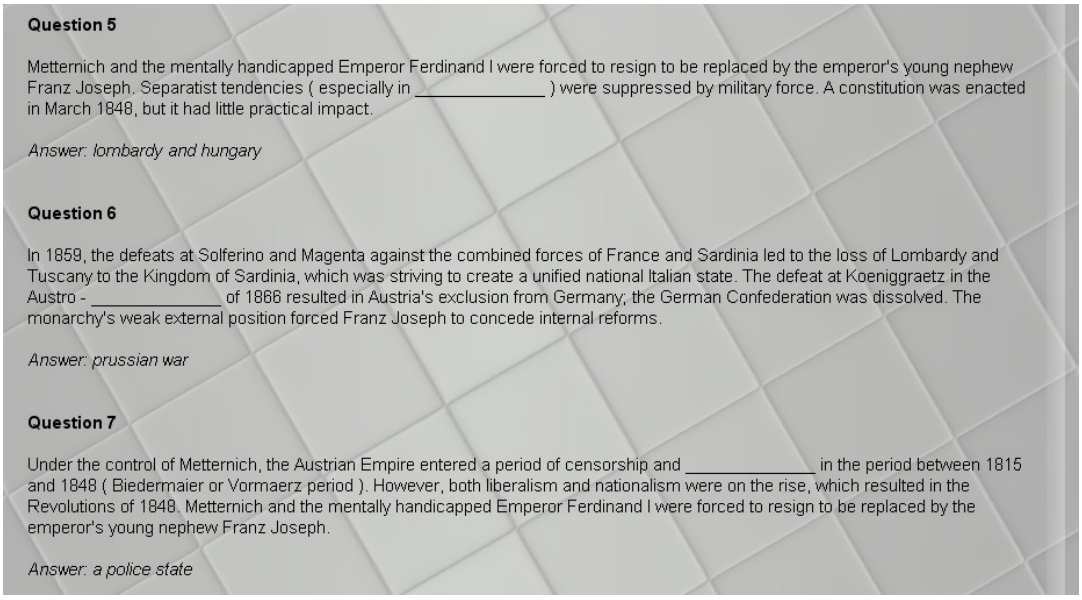

Abbildung 25: Erzeugte Lückentextfragen

7.4 Evaluierung

Die Evaluierung des Concept Extractors wurde als Teil der Evaluierung des Automatic Question Creators durchgeführt. Diese Evaluierung wurde gemeinsam mit Klaus Lankmayr (2010) durchgeführt, da sich die Evaluierung nur für das Gesamtsystem als zielführend herausgestellt hat. In diesem Abschnitt werden allerdings nur jene Ergebnisse der Evaluierung aufgezeigt, die mit dem Concept Extractor in Verbindung stehen. Dabei wurden vor allem die extrahierten Konzepte näher beleuchtet sowie die Benutzeroberfläche bewertet.

Prinzipielle Vorgehensweise

Die fünf ausgewählten Probanden waren Personen im Alter zwischen 24 und 30 Jahren, wobei eine Testperson davon weiblich war. Unter den Testpersonen befanden sich 4 Studenten und ein Softwareentwickler (mit HTL Abschluss und 10-jähriger Berufserfahrung). Von den vier Studenten studierten 2 Telematik, eine Person studiert Betriebswirtschaftslehre und die vierte Person studierte Pädagogik, Betriebswirtschaftslehre und Psychologie.

Das Beispielsdokument war dabei ein englischsprachiger, von Chris Hendrickson, verfasster Text, der über das Portal des MIT OpenCourseWare gefunden wurde und von Projektmanagement handelt (vgl. MIT OCW, 2010). Dabei wurden allerdings nur die ersten drei Abschnitte des zweiten Kapitels verwendet. Zusätzlich sei hierbei erwähnt dass diese Art der gesamten Evaluierung eine Ableitung der Evaluierung von Fragen mittels der Observation Matrix von Cannella, Ciancimino & Campos (2010) ist. Zu Beginn wurden die Probanden gebeten, das Beispielsdokument, aus welchen die Konzepte und Fragen extrahiert worden sind, aufmerksam zu lesen. Nach dem Lesen wurden sie aufgefordert die fünf wesentlichsten Konzepte (Wörter und Phrasen) der einzelnen Kapitel zu identifizieren. Anschließend wurden den Testpersonen die vom Concept Extractor extrahierten Konzepte zur Bewertung vorgelegt.

Nach dem bewerten der Konzepte wurden die Probanden gebeten, für jeden der unterstützten Fragetypen des Automatic Question Creator je zwei Fragen je Kapitel von Hand zu erzeugen. Danach wurde den Probanden der Evaluierungsbogen vorgelegt. Auf diesem befanden sich pro Kapitel und pro Fragetyp jeweils vier Fragen. Diese vier Fragen bestanden dabei jeweils aus zwei von Hand erzeugten und zwei vom Automatic Question Creator erzeugten Fragen in beliebiger Reihenfolge. Für die Evaluierung wurden die Probanden gebeten, die einzelnen Fragen nach bestimmten Kriterien zu bewerten. Diese Kriterien sind folgende:

1. *pertinence*: Dieser Wert gibt die Relevanz der Frage in Bezug auf das Thema des Textes bzw. des Abschnittes an.

2. *level*: Dieser Wert gibt den Schwierigkeitsgrad der Frage an.

3. *distractors*: Dieses Kriterium soll ein Maß für die Qualität der Distraktoren darstellen (nur bei Multiple Choice Fragen)

4. *concept*: Dieser Wert soll die Relevanz des extrahieren Konzeptes, welches die Grundlage für die erzeugten Fragen darstellt, angeben (nur bei Lückentextfragen, da sich das Konzept bei den unterschiedlichen Fragetypen nicht unterscheidet).

5. *answer*: Dieser Wert gibt die Qualität des berechneten Textabschnittes für die Referenzantwort bei Offenen Fragen an.

Auswertung

Für die Auswertung der Evaluierungsergebnisse werden vom gesamten Evaluierungsprozess nur die Werte für die Relevanz der einzelnen Konzepte (*concept*) bzw. die von den Probanden extrahierten Konzepte verwendet. Daher kann diese Auswertung in zwei Schritte unterteilt werden.

Als erstes werden die Extrahierten Konzepte der Probanden untereinander und im Vergleich zum Concept Extractor bewertet. Dabei werden drei Unterscheidungen getroffen. Einerseits werden die Übereinstimmungen der von den Probanden ermittelten Konzepte mit den fünf besten, vom Concept Extractor extrahierten, Konzepten bewertet, anderseits werden für diesen Vergleich die zehn besten Konzepte des Systems verwendet. Darüber hinaus werden die einzelnen Konzepte der Probanden untereinander verglichen. Bei der Auswertung wird die durchschnittlichen Übereinstimmungen der Konzepte in den einzelnen Kapiteln ermittelt und anschließend der Mittelwert über alle Kapitel berechnet. Dabei ist zu erwähnen, dass bei der Auswertung eine vollständige Übereinstimmung als Übereinstimmung klassifiziert wurde. Eine teilweise Übereinstimmung wurde hingegen als halbe Übereinstimmung bewertet. Dies resultiert aus der Beobachtung, dass die Probanden oftmals relativ lange Phrasen extrahiert haben, welche nur teilweise übereinstimmen. Die Ergebnisse sind in Abbildung 26 dargestellt.

Wie in dieser Abbildung ersichtlich zeigte sich, dass die Konzepte der Testpersonen untereinander zu ca. 31% übereinstimmen. Eine vollständige Übereinstimmung tritt dabei in 21% der Fälle auf. Die Übereinstimmungen der Konzepte der Testpersonen mit den fünf besten Konzepten des Concept Extractors treten bei ca. 22% der Fälle auf, für vollständige Übereinstimmungen trifft dies auf

rund 15% der Konzepte zu. Wenn nun die zehn besten Konzepte des Concept Extractors als Vergleichskonzepte verwendet werden liegt dieser Prozentsatz bei 27% und in Bezug auf vollständige Übereinstimmungen bei 17%. Daraus erkennt man, dass die Testpersonen etwas höhere Übereinstimmungen untereinander aufweisen, als im Vergleich mit dem System, allerdings lässt diese Analyse noch keine Rückschlüsse auf die Qualität der extrahierten Konzepte zu.

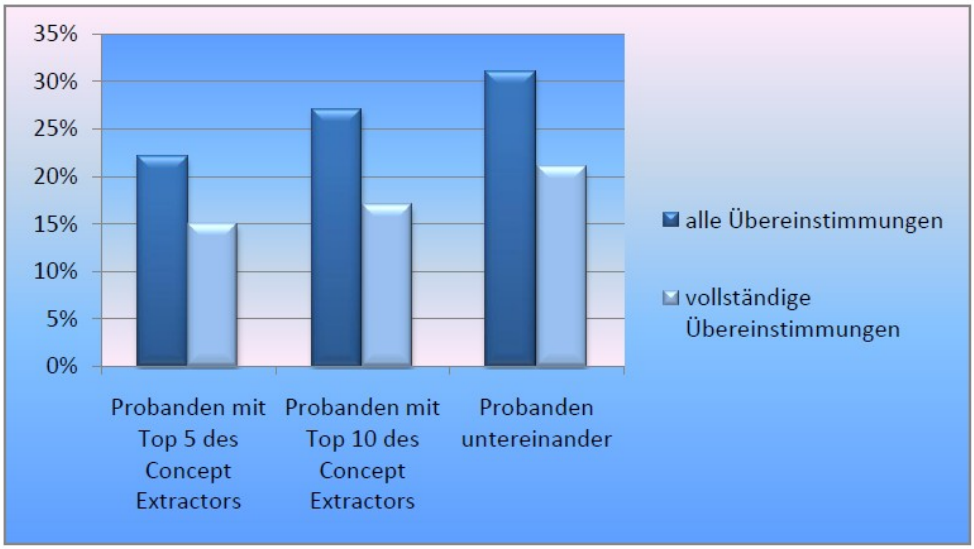

Abbildung 26: Übereinstimmungen der Konzepte

Im zweiten Schritt der Auswertung werden die Bewertungen der vom System extrahierten Konzepte evaluiert. Abbildung 27 zeigt den durchschnittlichen Wert der Relevanz der Konzepte für jedes Kapitel bzw. für das gesamte Dokument.

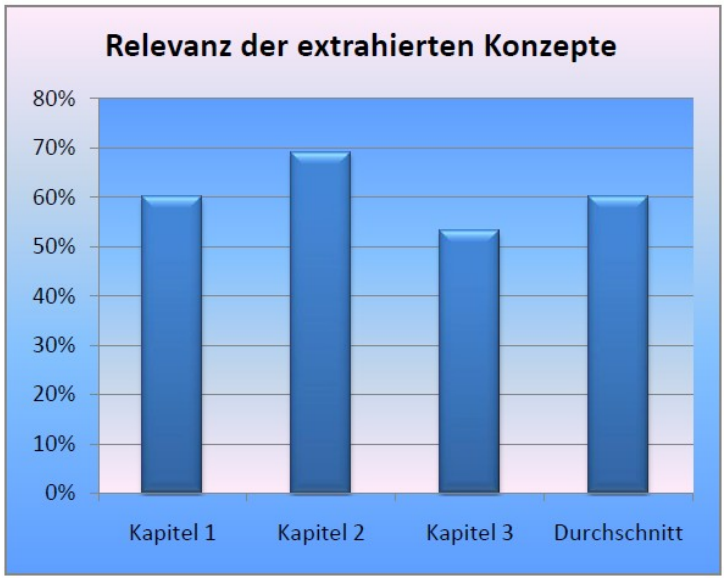

Abbildung 27: Vergleich der Relevanz der extrahierten Konzepte

Wie in dieser Abbildung 27 ersichtlich ist die durchschnittliche Relevanz der Konzepte in Kapitel zwei am höchsten. Dies kann eventuell damit begründet werden, dass dieses das längste Kapitel im Testdokument ist. Des Weiteren wurde ermittelt, dass von den dreißig vorgelegten Konzepten ca. 70% relevant sind. Ein Konzept wird dabei als relevant klassifiziert, wenn die durchschnittliche Relevanz eines Konzeptes einen Wert über 50 % aufweist (vgl. Abbildung 28).

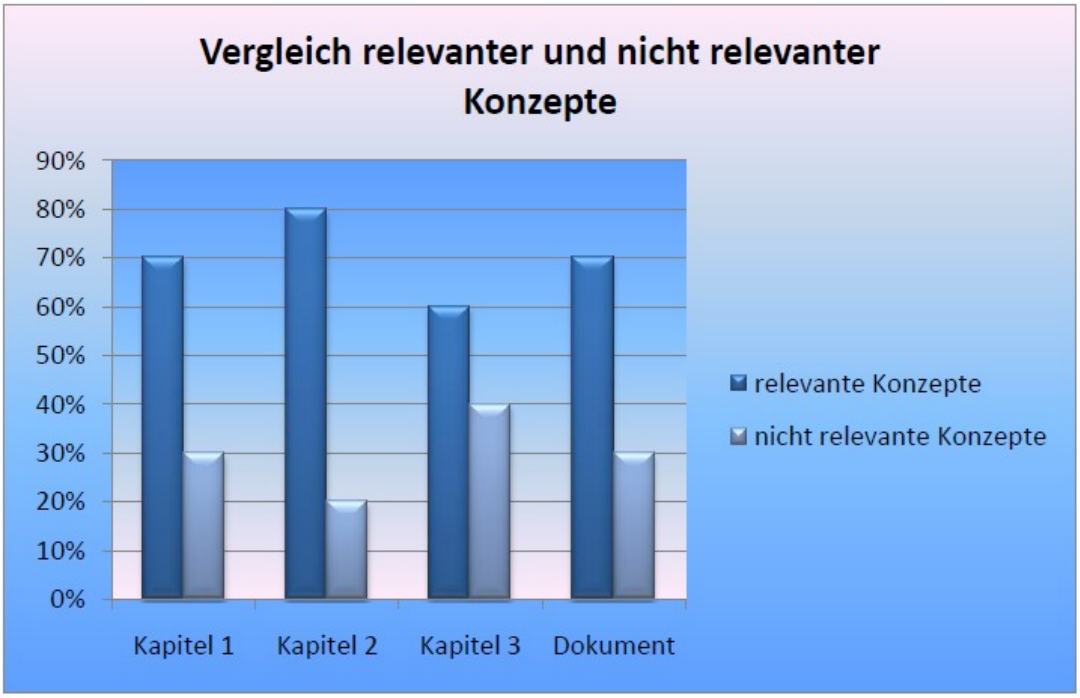

Abbildung 28: Vergleich von relevanten und nicht relevanten Konzepten

Diese Evaluierung verdeutlichte, dass der Concept Extractor durchaus in der Lage ist, zufriedenstellende Konzepte aus einem Dokument zu extrahieren. Die grafische Benutzeroberfläche und die Usability des Systems wurden von Seiten der Testpersonen durchwegs positiv bewertet. Sie fanden sich in diesem System rasch zurecht und lobten dabei die klare Strukturierung des Systems sowie die Vielzahl an Konfigurationsmöglichkeiten. Des Weiteren bewerteten die Probanden die ausführlich angezeigten Informationen in den einzelnen Prozessabschnitten als sehr positiv. Negativ erwähnt wurde hauptsächlich die relativ langen Wartezeiten, welche bei der Ausführung der einzelnen Berechnungsabschnitte auftreten. Diese können allerdings mit den vielfältigen und rechenintensiven Prozessabläufen erklärt werden.

7.5 Mögliche Erweiterungen und Verbesserungen

Im Zuge des Entwicklungsprozesses und der Testphase sind einige Probleme aufgetreten, welche die Ergebnisse des Systems negativ beeinflussen. Um diese Einflüsse zu vermindern, gibt es nun mehrere Erweiterungs - und Verbesserungsmöglichkeiten.

Eines der Hauptprobleme dieses Systems ist die Abhängigkeit von den Ergebnissen, die GATE und WordNet liefern. Wird von diesen Tools ein Term missinterpretiert, so zieht sich der Fehler meistens durch das gesamte Programm und beeinflusst das endgültige Ergebnis oftmals negativ. Um dies zu verhindern, ist es dankbar anderen NLP Tools zusätzlich zu integrieren, um die Ergebnisse zu verbessern und um dieses Tools zu unterstützen. Vor allem die Namenserkennung (sehr sprachspezifisch) und die Koreferenzauflösung lieferten oftmals schlechte Ergebnisse. Für Koreferenzen würde sich außerdem eine Integration in den Auswahlprozess dahingehend anbieten, dass auch die Koreferenzen als mögliche Konzepte ausgewählt werden, wobei diese natürlich bei der Anzeige durch die Wörter, die sie referenzieren, ersetzt werden müssten, da diese Phrasen sonst keinen Sinn ergeben würden.

Eine weitere Verbesserung würde mit relativ großer Wahrscheinlichkeit die Adaptierung des Abschnitterkennungssystems mit sich bringen. Im Moment findet ein Abschnittswechsel statt, wenn eine Überschrift der höchsten Ebene auftritt. Es wäre von Vorteil, wenn dabei auch die anderen Überschriftsebenen in geeigneter Art und Weise mit einbezogen werden. Dafür müsste dann eventuell auch die Berechnung des Überschriftengewichtsfaktors dahingehend adaptierte werden, so dass auch die unterschiedlichen Überschriftsebenen berücksichtigt werden.

Darüber hinaus könnte eine zusätzliche statistische Ermittlung des gemeinsamen Auftretens von Wörtern (Kollokationen bzw. Kookkurrenzen) zu besseren Ergebnissen führen. Des Weiteren wäre es von Vorteil, die Erkennung von Akronymen und Anaphern zu verbessern bzw. zu integrieren, um vor allem die statistischen Gewichte und damit auch die Endgewichte weiter verbessern zu können.

Eine Wortbedeutungsdisambiguierung würde ebenfalls einen Vorteil mit sich bringen. Wenn die genaue Bedeutung eines Wortes bekannt ist, so wird die Auswahl der ähnlichen Wörter aller Voraussicht nach genauer und damit erheblich verbessert.

Ein erheblicher Nachteil des Systems ist natürlich die sehr eingeschränkte Verarbeitung von PDF Dateien. Daher bietet es sich auch hierbei an, den PDFtoHTML Konverter zu verbessern bzw. einen anderen Konverter zu integrieren, welcher solche Dateien besser unterstützt und es somit ermöglicht, alle benötigten Informationen aus den Dokumenten zu extrahieren.

Ein weiterer Verbesserungsvorschlag ist die Adaptierung des Auswahlsystems der relevanten Wörter. Im Moment werden alle Wörter, deren Endgewicht über einen bestimmten Wert liegt, als relevant klassifiziert. Es wäre wahrscheinlich vorteilhaft, in dieses Auswahlverfahren in geeigneter Art und Weise einzugreifen,

um auf die extrahierten Wörter und Phrasen besser Einfluss nehmen und auch die Anzahl der extrahierten Wörter direkt bestimmen zu können

Die Verwendung von themenspezifischen Korpora würde die Performance des Systems aller Voraussicht nach ebenfalls erheblich verbessern. Dabei würde einerseits das System beschleunigt, andererseits würde sich die Auswahl der relevanten Wörter und Phrasen dahingehend verbessern, dass dann gezielt jene Wörter ausgewählt werden könnten, die durch den Korpus für dieses Thema als relevant identifiziert werden können. Die Verwendung von Korpora bringt allerdings auch erhebliche Nachteile mit sich. Einerseits wäre die Verarbeitung nur mehr Themenspezifisch durchführbar, d. h. dass nur mehr Texte aus jenen Themen analysiert werden könnten, für welche ein themenspezifischer Korpus existiert. Zusätzlich ist die Erstellung eines solchen Korpus darüber hinaus ein sehr aufwendiger Prozess und das System daher eventuell nicht mehr zielführend.

Ebenfalls zielführend wäre sicherlich eine Optimierung der Implementierten Algorithmen, da diese relativ rechenintensiv sind und daher beispielsweise der Einsatz als Webservice nur eingeschränkt möglich ist. Dabei ist allerdings zu erwähnen, dass die Rechenzeit sehr stark von der Länge des Dokuments bzw. von der Länge der einzelnen Abschnitte abhängig ist.

7.6 Zusammenfassung

In diesem Kapitel wurde der Concept Extractor, ein System zur Extraktion von semantisch relevanten Daten aus natürlich sprachlichen Inhalten mit nachfolgender automatischer Fragengenerierung, vorgestellt. Dieser Concept Extractor ist Teil des Automatic Question Creators, der mittels der extrahierten Konzepte automatisch Fragen erzeugt. Konkret wurden in diesem Kapitel die prinzipielle Architektur aufgezeigt, die Implementierung erläutert und die konkrete Applikation vorgestellt.

Das System arbeitet prinzipiell in vier Schritten. Im ersten Schritt wird ein beliebiges Eingangsdokument in ein internes Repräsentationschema umgewandelt, welches alle wichtigen Informationen für die weitere Verarbeitung beinhaltet. Danach findet eine Textvorverarbeitung statt, wobei dem Text mit Hilfe der beiden NLP Tools GATE und WordNet spezielle für die weitere Verarbeitung benötigte Informationen hinzugefügt werden.

Anschließend findet die Textanalyse statt. Dabei werden statistische, strukturelle und semantische Analysen durchgeführt. Die bei diesen Analysen ermittelten Eigenschaften und Merkmale werden anhand verschiedenster Methoden für eine Berechnung von Gewichten für jedes einzelne Wort herangezogen. Das resultierende Gewicht eines Wortes gibt abschließend Aufschluss über die Relevanz des Wortes in Bezug auf den Inhalt, d. h. je höher das Gewicht, desto besser ist das

Wort geeignet, den Inhalt zu repräsentieren. Mittels der mit dieser Methode bestimmten Wörter werden dann abschließend Phrasen gebildet und diese dem Benutzer zugänglich gemacht.

Diese Wörter und Phrasen dienen dem Automatic Question Creator in weiterer Folge als Grundlage für die automatische Fragengenerierung. Dieser Prozess ist allerdings nicht Teil dieser Arbeit und deshalb sei an dieser Stelle nochmals auf die Masterarbeit *Entwurf und Entwicklung von Konzepten für die automatische Fragengenerierung* von Klaus Lankmayr (2010) verwiesen.

Zusätzlich wurde die grafische Benutzeroberfläche des Systems aus der Sichtweise eines Nutzers vorgestellt. Dabei wurde auf den grundsätzlichen Aufbau dieser Oberfläche näher eingegangen sowie die Funktionalität, die dieses Programm bietet, erläutert. Darüber hinaus wurden die Konfigurationsmöglichkeiten des Systems aufgezeigt, sowie anhand eines Beispiels die angezeigten Ergebnisse interpretiert.

Der darauf folgende Abschnitt dieses Kapitels befasste sich mit einer Evaluierung des Systems, wobei die Ergebnisse auf ihre Sinnhaftigkeit überprüft wurden. Dabei zeigte sich, dass ungefähr 70% der vom Concept Extractor extrahierten Konzepte als relevant anzusehen sind. Die durchschnittliche Relevanz der Konzepte beträgt ca. 60 %. Darüber hinaus wurde aufgezeigt, dass die Anzahl der Übereinstimmungen zwischen den vom System extrahierten Konzepte und den von den Testpersonen extrahierten Konzepten vergleichbar ist mit der Anzahl der Übereinstimmungen, welche die Konzepte der Probanden untereinander aufweisen.

Der abschließende Teil dieses Kapitels beinhaltet mögliche Erweiterungs – und Verbesserungsvorschläge für das System. Dabei wurde auf die Möglichkeiten, eventuell andere, zuverlässigere externe Tools in das System zu integrieren, eingegangen, sowie Adaptierungen der bestehenden Algorithmen vorgeschlagen. Zusätzlich wurden andere Analysemethoden, wie beispielsweise der Einsatz von Korpora oder die Verarbeitung von Kollokationen bzw. Kookkurrenzen, aufgezeigt.

Die Implementierung dieses Systems und die vorangehenden Recherchen waren dem Autor in vielerlei Hinsicht lehrreich und informativ. Die gewonnen Erkenntnisse und Erfahrungen werden daher im nachfolgenden Kapitel erläutert.

8 Lessons learned

Die Erstellung dieser Arbeit und die Implementierung des Concept Extractors war für den Autor ein sehr intensiver und arbeitsaufwendiger, aber auch ein sehr lehrreicher Prozess. Die dabei gewonnenen Erfahrungen und erlangten Erkenntnisse werden sicherlich im weiteren Leben des Autors sehr hilfreich sein.

Die wichtigste Erkenntnis, welche in dieser Zeit gewonnen wurde, ist die Wichtigkeit einer detaillierten Planung und das damit einhergehende Entdecken von möglichen Problemen, bevor diese überhaupt auftreten, um bereits früh in geeigneter Art und Weise auf diese Probleme regieren zu können. Obgleich es natürlich oftmals sehr schwierig ist, diese Probleme in einer frühen Phase zu entdecken. Allerdings erkannte der Autor auch, dass selten alles planmäßig abläuft. Vor allem die Vorababschätzung der benötigten Zeit der einzelnen Arbeitsschritte funktionierte meistens weniger gut, was wahrscheinlich auch auf Erfahrungsmangel des Autors zurückzuführen ist. Allerdings ist anzunehmen, dass diese Abschätzungen in Zukunft aufgrund der gewonnen Erfahrungen bei diesem Projekt sicherlich zuverlässiger funktionieren werden.

Im Zuge der Literaturrecherchen erkannte der Autor, dass ein Großteil der Ressourcen nur in elektronischer Form verfügbar ist. Dabei ist anzumerken, dass diese elektronischen Ressourcen in sehr großer Anzahl vorhanden sind und es daher sehr schwierig ist, die am besten geeigneten Materialien herauszufiltern. Des Weiteren lernte der Autor, dass viele dieser Ressourcen strengen Restriktionen in Bezug auf Form und Länge des Textes unterliegen, sodass das zentrale Thema des Textes oftmals nur sehr eingeschränkt und wenig detailreich abgehandelt wird. Dadurch ist es sehr schwierig konkrete Information über die in diesen Arbeiten vorgestellten Methoden zu erhalten, um daraus geeignete Schlüsse zu ziehen und diese dann in den eigenen Entwurf mit einfließen lassen zu können.

Bei der Implementierung lernt der Autor gute und ausführliche Dokumentation von Software zu schätzen, da es große Probleme bereitete, die unterschiedlichsten Tools und Frameworks in das System zu integrieren, ohne die anderen Module dabei zu beeinflussen. Diese Schwierigkeiten waren oftmals auch auf das Fehlen einer geeigneten Dokumentation zurückzuführen. Leider machte der Autor die Erfahrung, dass sehr viele Software Projekte nicht bzw. nur sehr rudimentär dokumentiert sind und sich daher teilweise die Verwendung dieser als nahezu unmöglich herausstellte.

Des Weiteren bekam der Autor einen Einblick in die Komplexität der Textverarbeitung. Es zeigte sich, dass bereits oftmals unbedeutende Zeichen und Symbole schwerwiegende Konsequenzen für die korrekte funktionsweise einer Applikation haben können. Dabei wurde auch erkannt, dass es relativ schwierig ist,

aus dem Konglomerat der unterschiedlichen Dateiformate ein allgemeines Format zu erzeugen, welches alle gewünschten bzw. nur die gewünschten Informationen beinhaltet.

Eine weitere Erfahrung die der Autor in der Implementierungsphase machte war jene, dass es oftmals besser ist, nicht von externen Tools und Frameworks abhängig zu sein, da oftmals viele Kompromisse eingegangen werden müssen, aufgrund dessen, dass diese selten genau jene Funktionalität anbieten, die tatsächlich benötigt wird. Dabei erwies es sich oft als vorteilhaft, die gewünschte Funktionalität selbst zu implementieren und damit einhergehend keine Kompromisse bezüglich der Funktionsweise eingehen zu müssen.

Trotz, oder vielleicht sogar auf Grund, der oftmals negativen Erfahrungen, welche der Autor im Zuge des Projektes gemacht hat, sei hierbei allerdings erwähnt, dass die Realisierung des Concept Extractors und dieser Masterarbeit außerordentlich viel Spaß bereitet hat und darüber hinaus sehr lehrreich, informativ und horizonterweiternd war.

9 Zusammenfassung

Das Ziel dieser Arbeit war es, ein System zu entwerfen, welches aus natürlich sprachlichen Texten jene Konzepte, also Wörter und Phrasen extrahiert, die den wesentlichen Inhalt und die zentralen Themen des Textes repräsentieren. Zusätzlich sollen aufbauend auf diesen Konzepten automatisch Fragen erzeugt werden. Diese Fragen sollen dabei eine Überprüfung des Verständnisses des Ausgangstextes ermöglichen.

Zu Beginn dieser Arbeit wurden grundlegende Untersuchungen von Sprache und Text durchgeführt und in die Verarbeitung natürlicher Sprache eingeführt. Zusätzlich wurden die Teilgebiete der Verarbeitung natürlicher Sprache, welche mit dem Thema dieser Arbeit in enger Beziehung stehen, näher erläutert und erste Erkenntnisse daraus extrahiert. So können die Wörter prinzipiell in zwei Klassen unterteilt werden, wobei prinzipiell nur die Klasse der Inhaltswörter für die Bedeutung des Textes wichtig sind. Des Weiteren zeigte sich, dass die Anzahl der Vorkommen der einzelnen Wörter Rückschlüsse auf die Wichtigkeit des Wortes in Bezug auf den Inhalt zulässt. Dies führt in weiterer Folge zur statistischen Analyse der Texte, welche grundsätzlich dem Zählen der Wörter gleichzusetzten ist.

Für solche Textanalysen sind allerdings wichtige Vorverarbeitungsschritte unbedingt erforderlich. Als erstes ist es wichtig, den Text zu tokenisieren und die einzelnen Sätze im Text zu identifizieren. Zusätzlich wird für eine statistische Analyse eine Grundformreduktion benötigt, um die einzelnen Flexionsformen der Wörter als ein und dasselbe Wort identifizieren zu können. Diese Grundformreduktion erweist sich in weiterer Folge in Bezug auf die Ergebnisse der statistischen Analyse und auch in Bezug auf die weiteren Analysen als durchaus vorteilhaft. Bei weiteren Recherchen im Gebiet der natürlichen Sprachverarbeitung zeigte sich, dass die Analyse weiterer sprachrelevanter Eigenschaften für die Funktionalität der Extraktion von semantisch relevanten Daten durchaus zielführend ist. So wurde erkannt, dass Phrasen, in diesem Falle Nominalphrasen, eine höhere Aussagekraft über den Inhalt besitzen als die einzelnen Wörter. Daher ist es wichtig, diese Phrasen in den Texten zu identifizieren und sie im Auswahlprozess zu berücksichtigen. Zusätzlich zeigte sich, dass das Erkennen und das Auflösen von Akronymen, anaphorischen Ausdrücken und von Koreferenzen wesentlich zur Verbesserung der erzeilten Ergebnisse beitragen können. Für eine sinnvolle Interpretation eines Textes ist es ebenfalls von Bedeutung, Namen von Personen, Orten, Organisationen usw., zu erkennen, da diese oftmals zentrale Elemente in einem Text darstellen. Daher ist es zielführend, eine Eigennamenerkennung in den Analyseprozess zu integrieren.

Bei der Betrachtung des aktuellen Forschungsstandes zeigte sich rasch, dass eine sinnvolle Extraktion der relevanten Daten aus natürlich sprachlichen Texten nur

durch zusätzliche semantische Analysen durchführbar ist. Diese semantischen Analysen dienen dazu, die einzelnen Wörter zu untersuchen und ihre Verbindungen zueinander aufzuzeigen um daraus geeignete Schlüsse ziehen zu können. Dadurch ist es beispielsweise möglich, eine Wortbedeutungsdisambiguierung durchzuführen.

Des Weiteren zeigte sich diesen Recherchen, dass strukturelle Eigenschaften der Texte wichtige zusätzliche Erkenntnisse über den Inhalt liefern können. So geben beispielsweise die Titel, Überschriften, Kurzfassung, Schlüsselwörter usw. einen direkten Hinweis auf den näheren Inhalt des Textes. Zusätzlich wurde in den Untersuchungen oftmals auf die Bedeutung der Position der Wörter im Text bzw. in den Absätzen hingewiesen. Dieser Umstand konnte allerdings in weitere Folge bei weiteren Nachforschungen nicht bestätigt werden, da diese Eigenschaft zu sehr von der Art des Textes abhängt und daher nicht verallgemeinert werden kann.

Mittels dieser Erkenntnisse wurde anschließend der Concept Extractor entwickelt. Dieses Applikation soll die wesentlichen Konzepte aus Texten extrahieren und anschließend mit diesen Wörtern und Phrasen als zentrale Elemente Fragen generieren. Der prinzipielle Ablauf einer Textanalyse gliedert sich dabei in die Vorverarbeitung, die Textanalyse, die Bestimmung und die Extraktion der relevanten Konzepte und in die Fragengenerierung.

Bei der Vorverarbeitung wird dabei der Text aus einem beliebigen unterstützten Eingangsformat in ein internes Dateiformat umgewandelt. Dabei traten allerdings bereits die ersten Schwierigkeiten auf. Andererseits ist die Extraktion von benötigten Eigenschaften der Texte nicht immer einfach (vor allem bei PDF Dateien), andererseits befinden sich in den Dokumenten oftmals spezielle Zeichenfolgen, die die Funktionsweise des Systems erheblich stören. Der zweite Schritt bei der Vorverarbeitung ist die Textvorverarbeitung, wobei die grundlegenden, für die weitere Verarbeitung benötigten, syntaktischen und morphologischen Analysen durchgeführt werden. Dabei werden eine Tokenisierung, eine Satzgrenzenerkennung, eine Wortartenerkennung, eine Eigennamen-erkennung, ein Stemming sowie eine Koreferenzauflösung abgearbeitet.

Anschließend werden die Textanalysen durchgeführt. Die Textanalyse unterteilt sich dabei in statistische, strukturelle und semantische Analysen. Die dabei ermittelten Eigenschaften werden in geeignete numerische Werte umgewandelt und abschließend zu einem Gesamtgewicht für jedes Wort kombiniert. Bei der statistischen Analyse wird die Anzahl der Vorkommen der auf die Grundform reduzierten Wörter ermittelt. Die strukturellen Analyse berücksichtigt bestimmte Eigenschaften wie Formatierungen und spezielle Überschriften sowie bestimmte Annotationen von Wörtern, wie beispielsweise Personen, Orte, Organisationen usw. Bei der semantischen Analyse werden die semantischen und lexikalischen Verbindungen der einzelnen Wörter zueinander untersucht. Dabei werden im ersten Schritt für jedes Wort die ähnlichen Worte ermittelt. Die dabei berechnete

Ähnlichkeit und das statistische Gewicht der ähnlichen Wörter wird in die Berechnung des Endgewichtes des Ausgangswortes mit einbezogen. Anschließend werden die Einflüsse von Titel, Überschriften, Kurzfassung und Schlüsselwörtern auf das Endgewicht berechnet. Dabei wird für jedes Wort die Relevanz zu diesen Textpassagen ermittelt und anhand von bestimmten Faktoren die Auswirkung auf das Gesamtgewicht eines Wortes bestimmt. Diese Faktoren der einzelnen Analyseverfahren können darüber hinaus vom Benutzer beliebig kombiniert und individuell bestimmt werden.

Die einzelnen in den verschiedenen Analyseverfahren ermittelten Einflüsse auf das Endgewicht werden nach Abschluss der Analysen geeignet kombiniert und ein Gesamtgewicht für jedes Wort ermittelt. Anschließend werden die am höchsten gewichteten Worte für jeden Abschnitt extrahiert und mit diesen Wörtern geeignete Phrasen gebildet. Diese werden dem Benutzer in geeigneter Art und Weise angezeigt, worauf dieser die Möglichkeit hat, sich auf diesen Phrasen basierende Fragen erzeugen zu lassen.

Die Evaluierung des Systems verdeutlichte, dass der Concept Extractor außerordentlich zufriedenstellende Ergebnisse liefert. Es zeigte sich, dass durchschnittlich 70% der vom System extrahierten Konzepte als relevant angesehen werden. Die durchschnittliche Relevanz der einzelnen Konzepte liegt bei ungefähr 60%. Diese Werte veranschaulichen, dass mit dem Concept Extractor ein System entwickelt wurde, welches für weitere Forschungsarbeiten in diesem Gebiet als ausgezeichnete Grundlage dienen kann.

10 Literaturverzeichnis

Ahmad, K., & Gillam, L. (2005). Automatic Ontology Extraction form Unstructured Texts. In Meersman, R., & Tari, Z. (Hrsg.), *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE* (Seiten 1330-1246). Heidelberg: Springer Verlag.

Amtrup, J. W. (2001). Aspekte der Computerlinguistik. In Carstensen, K. U., Jekat, S., & Klabunde, R.,. Computerlinguistik – Was ist das? In Carstensen, K. U., Ebert, C., Endriss, C., Jekat, S., Klabunde, R., & Langer, H. (Hrsg.), *ComputerLinguistik und Sprachtechnologie: eine Einführung (*Seiten 1 – 25). Heidelberg: Spektrum Akademischer Verlag.

Apache (2010). Apache PDFBox: Java PDF Library. URL - http://pdfbox.apache.org/, (The Apache Software Foundation), (Zugriffsdatum: 20.Februar 2010).

Art of Solving (2010). JOD Converter. URL - http://artofsolving.com/opensource/jodconverter, (Zugriffsdatum: 20. Februar 2010).

Baader, F., Horrocks, I., & Sattler, U. (2004). Description Logics. In Staab, S. & Studer, R. (Hrsg.), *Handbook on Ontologies (International Handbooks on Information Systems)* (Seiten 3 – 28). Heidelberg: Springer Verlag.

Baeza - Yates, R., & Ribeiro - Neto, B. (1999). *Modern Information Retrieval*. New York: ACM Press.

Barzilay, R., & Elhadad, M. (1999). Using Lexical Chains for Text Summarization. In: Mani, I., Maybury,M.T. (Hrsg.), Advances in Automatic Text Summarization (Seiten 111–121). Cambridge : MIT Press.

Bawakid, A. & Oussalah, M. (2008). A Semantic Summarization System: University of Birgmingham at TAC 2008. In Proceedings of the First Text Analysis Conference (TAC 2008) (Gaithersburg, Maryland, USA). URL - http://www.nist.gov/tac/publications/2008/index.html (Zugriffsdatum: 6. Februar 2010).

Beeferman, D., Berger, A., & Lafferty J. (1999). Statistical Models for Text Segmentation. *In Machine Learning, 34* (1-3), Seiten 177 -210. Dordrecht: Kluwer Academic Publishers.

Bertelsmann (1996a). Die große Bertelsmann Lexikothek: Bertelsmann Lexikon (Band 13, Seiten 335 - 336). Gütersloh: Bertelsmann Lexikothek Verlag.

Bertelsmann (1996b). Die große Bertelsmann Lexikothek: Bertelsmann Lexikon (Band 13, Seiten 101 - 104). Gütersloh: Bertelsmann Lexikothek Verlag.

Bertelsmann (1996c). Die große Bertelsmann Lexikothek: Bertelsmann Lexikon (Band 12, Seite 131). Gütersloh: Bertelsmann Lexikothek Verlag.

Bertelsmann (1996d). Die große Bertelsmann Lexikothek: Bertelsmann Lexikon (Band 13, Seite 184). Gütersloh: Bertelsmann Lexikothek Verlag.

Brückner, T. (2001). Textklassifikation. In Carstensen, K. U., Anwendungen. In Carstensen, K. U., Ebert, C., Endriss, C., Jekat, S., Klabunde, R., & Langer, H. (Hrsg.), *Computerlinguistik und Sprachtechnologie: eine Einführung* (Seiten 514 - 521). Heidelberg: Spektrum Akademischer Verlag.

Cannella, S., Ciancimino, E., & Campos M. L. (2010). *Mixed e-Assessment: an application of the student-generated question technique*. IEEE Engineering Education 2010 – The Future of Global Learning in Engineering Education.

Carstensen, K. U. Jekat, S., & Klabunde, R. (2001). Computerlinguistik – Was ist das? In Ebert, C., Endriss, C., Methoden. In Carstensen, K. U., Ebert, C., Endriss, C., Jekat, S., Klabunde, R., & Langer, H. (Hrsg.), *Computerlinguistik und Sprachtechnologie: eine Einführung* (Seiten 1 – 23). Heidelberg: Spektrum Akademischer Verlag.

Carstensen, K. U. (2001). Anwendungen. In Carstensen, K. U., Ebert, C., Endriss, C., Jekat, S., Klabunde, R., & Langer, H. (Hrsg.), *Computerlinguistik und Sprachtechnologie: eine Einführung* (Seiten 409 – 521). Heidelberg: Spektrum Akademischer Verlag.

Choi, F. Y. Y. (2000). Advances in domain independent linear text segmentation. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, Seiten 26-33, San Francisco: Morgan Kaufmann Publishers Inc.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms* (2. überarbeitete Ausgabe). Cambridge: The MIT Press.

Cruz, C. M., & Urrea, A. M. (2005). Extractive Summarization Based on Word Information ans Sentence Position. In Gelbukh, A. (Hrsg), *Computational Linguistics and Intelligent Text Processing, Proceedings of 6th International*

Conference, CICLing 2005 (Mexiko City, Mexiko, 2005), Seiten 653 - 656. Heidelberg: Springer Verlag.

Cunningham, H. (2002). Gate, a General Architecture for Text Engineering. Computer and the Humanities, 36 (2), Seiten 223 – 254. Dordrecht: Kluwer Academic Publishers.

Cunningham, H., Maynard, D., Bontcheva, K., & Tablan, V. (2002). GATE: an Architecture for Development of Robust HLT Applications. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (Philadelphia, Pennsylvania, 2002), Seiten 168 -175.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Dimitrov, M., Dowman, … Roberts, A. (2010). Developing Language Processing Components with GATE Version 5 (a User Guide).

Dorna, M. (2001). Maschinelle Übersetzung. In Carstensen, K. U., Anwendungen. In Carstensen, K. U., Ebert, C., Endriss, C., Jekat, S., Klabunde, R., & Langer, H. (Hrsg.), *Computerlinguistik und Sprachtechnologie: eine Einführung* (Seiten 514 - 521). Heidelberg: Spektrum Akademischer Verlag.

Edmundson, H. (1969). New methods in automatic extracting. In: Journal oft h ACM, 16 (2), Seiten 264 -285. New York: ACM Press.

Endress-Niggemeyer, B. (2001). Textzusammenfassung. In Carstensen, K. U., Anwendungen. In Carstensen, K. U., Ebert, C., Endriss, C., Jekat, S., Klabunde, R., & Langer, H. (Hrsg.), *Computerlinguistik und Sprachtechnologie: eine Einführung* (Seiten 409 – 521). Heidelberg: Spektrum Akademischer Verlag.

Erkan, G., & Cicekli, I. (2008). Lexical Cohesion Based Topic Modeling for Summarization. In Gelbukh, A. (Hrsg), *Computational Linguistics and Intelligent Text Processing, Proceedings of 9th International Conference, CICLing 2008* (Haifa, Israel, 2008), Seiten 582 - 592. Heidelberg: Springer Verlag.

Evert, S., & Fitschen, A. (2001). Textkorpora. In Carstensen, K. U., Ressourcen. In Carstensen, K. U., Ebert, C., Endriss, C., Jekat, S., Klabunde, R., & Langer, H. (Hrsg.), *Computerlinguistik und Sprachtechnologie: eine Einführung* (Seiten 369 - 376). Heidelberg: Spektrum Akademischer Verlag.

Fellbaum, C. (1998a). Introduction. In Fellbaum, C. (Hrsg.), *WordNet: An Electronic Lexical Database,* (2. Druck) (Seiten 1 - 19). Cambridge: MIT Press.

Fellbaum, C. (1998b). A Semantic Network of English Verbs. In Fellbaum, C. (Hrsg.), *WordNet: An Electronic Lexical Database,* (2. Druck) (Seiten 1 - 19). Cambridge: MIT Press.

Ferret, O. (2002). Using collocations for topic segmentation and link detection. In Proceedings of the 19th international conference on Computational linguistics, (Taipei, Taiwan), Volume 1, Seiten 1-7, Morristown: Association for Computational Linguistics.

Gansel, C., & Jürgens, F. (2007). *Textlinguistik und Textgrammatik: Eine Einführung* (2. Auflage). Göttingen : Vandenhoeck & Ruprecht.

Galley, M., & McKeown, K. (2003) Improving word sense disambiguation in lexical chaining. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence* (Acapulco, Mexiko, 2003), Seiten 1486–1488. San Francisco: Morgan Kaufmann Publishers Inc.

GATE (2010). GATE: general architecture for text engineering. URL - http://gate.ac.uk/, (Zugriffsdatum: 18 Februar 2010).

Gelfand, B., Wulfekuhler, M., & Punch, W. F., III (1998). Automated Concept Extraction From Plain Text. *In Papers from the AAAI 1998 Workshop on Text Categorization*, Seiten 13 – 17.

Grossman, D. A., & Frieder, O. (2004). Information Retrieval: Algorithms and Heuristics (2. Ausgabe). Dordrecht: Springer Verlag.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge Acquiition, 5(2), Seiten 199-220. London: Academic Press Ltd.

Guardian (2010). Guardian: Internet data heads for 500bn gigabytes. URL - http://www.guardian.co.uk/business/2009/may/18/digital-content-expansion. (Zugriffsdatum: 23. Jänner 2010).

HaCohen-Kerner, Y., Gross, Z., & Masa, A. (2005). Automatic Extraction and Learning of Keyphrases from Scientific Articles. In Gelbukh, A. (Hrsg), *Computational Linguistics and Intelligent Text Processing, Proceedings of 6th International Conference, CICLing 2005* (Mexiko City, Mexiko, 2005), Seiten 657 - 669. Heidelberg: Springer Verlag.

Hausser, R. (2000). *Grundlagen der Computerlinguistik: Mensch-Maschine-Kommunikation in natürlicher Sprache*. Heidelberg: Springer Verlag

Haenelt, K. (2009). Fraunhofer: Information Retrieval Modelle: Boolesches Modell. URL - http://kontext.fraunhofer.de/haenelt/kurs/folien/Haenelt_IR_Modelle_Boole.pdf, (Zugriffsdatum: 29.Jäner 2010).

Hearst, M. A., & Plaunt, C. (1993). Subtopic Structuring for Full-Length Document Access. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval* (Pittsburgh, Pennsylvania, United States), Seiten 59 – 68. New York: ACM Press.

Hearst, M. A. (1997). TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages, *Computiational Linguistics, 23* (1), Seiten 33 – 64. Cambridge: MIT Press.

Hesse, W. (2002). Ontologie(n). In *Informatik-Spektrum*, 25 (6), Seiten 477 - 480. Heidelberg: Springer Verlag.

HTML Cleaner (2010). URL - http://htmlcleaner.sourceforge.net/, (Zugriffsdatum: 20.Februar 2010).

Hutchinson, J. W., & Sommers, H. L. (1992). *An Introduction to Machine Translation*. San Diego: Academic Press Inc.

Jackson, P., & Moulinier, I. (2002). *Natural Language Processing for online Applications: Text Retrieval, Extraction and Categorization*. Amsterdam: John Benjamins Publishing.

JDOM (2010). JDOM. URL - http://www.jdom.org/index.html, (Zugriffsdatum: 20.Februar 2010).

Jiang, J. J. and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *International Conference Research on Computational Linguistics*.

Jurafsky, D., & Martin, J. H. (2008). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (2. Auflage). Upper Saddle River: Prentice Hall.

Kof, L., & Pizka, M. (2005). Validating Documentation with Domain Ontologies. In Fujita, H., & Mejri, M. (Hrsg.), *New trends in software methodologies, tools and techniques: proceedings of the fourth SoMeTW 05*, Seiten 126 -143. Amsterdam: IOS Press.

Kunze, C. (2001). Lexikalisch-semantische Netze. In Carstensen, K. U., Ressourcen. In Carstensen, K. U., Ebert, C., Endriss, C., Jekat, S., Klabunde, R., & Langer, H. (Hrsg.), *Computerlinguistik und Sprachtechnologie: eine Einführung* (Seiten 369 - 376). Heidelberg: Spektrum Akademischer Verlag.

Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. In *SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, Seiten 68-73, New York, NY, USA. ACM.

Kürschner, W. (2007). *Taschenbuch Linguistik*: Ein Studienbegleiter für Germanisten (3. durchgesehene Auflage). Berlin: Erich Schmid Verlag.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). Introduction to Latent Semantic Analysis. Discourse Processes, 24, Seiten 259-284.

Lankmayr, K. (2010). *Entwurf und Entwicklung von Konzepten zur automatischen Fragengenerierung* (Masterarbeit, TU Graz, IICM, 2010).

Ledeneva, Y., Gelbukh, A., & García - Hernández, R. A. (2008). Terms Derived from Frequent Sequences for Extractive Text Summarization. In Gelbukh, A. (Hrsg), *Computational Linguistics and Intelligent Text Processing, Proceedings of 9th International Conference, CICLing 2008* (Haifa, Israel, 2008), Seiten 593 - 604. Heidelberg: Springer Verlag.

Lehmnitzer, L., & Zinsmeister H. (2006). *Korpuslinguistik: Eine Einführung.* Tübingen: Narr Verlag.

Lehner, F. (2008). *Wissensmanagement: Grundlagen, Methoden und Unterstützung*, 2. Auflage. München: Carl Hanser Verlag.

Lehrberger, J. (2003). Automatic Translation and the Concept of Sublanguage. In Nierenburg, S., Sommers, H. L., & Wilks, Y. (Hrsg.), *Readings in machine translation* (illustrierte Ausgabe, Seiten 207 - 221). Cambridge: MIT Press.

Lewis, D. D. (1992). Naive (Bayes) at forty: The independence assumption in information retrieval. *In Proceedings of ECML-98, 10th European Conference on Machine Learning* (Chemnitz, Germany, 1998), Seiten 4–15. Heidelberg: Springer Verlag.

Lin, D. (1998). An information-theoretic definition of similarity. In Shavlik, J. W. and Shavlik, J. W., (Hrsg.), *ICML* (Seiten 296-304). San Francisco: Morgan Kaufmann Publishers.

Lin, C.-Y. (1995). Knowledge-based automatic topic identification. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics* (Cambridge, Massachusetts, 1995).

Lippe, W.-M., (2005). *Soft-Computing: Mit neuronalen Netzen, Fuzzy-Logic und evolutionären Algorithmen*. Heidelberg: Springer Verlag.

Liu, B. (2009). *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*, 2. korrigierte Ausgabe. Heidelberg: Springer Verlag.

Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. *IBM Journal of Research, 2* (4), Seiten 159 – 164.

Mani, I. (2001). *Automatic Summarization (Natural Language Processing, 3 (Paper))*. Amsterdam: John Benjamins Publishing Co.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge: Cambridge University Press.

Manning, C. D., & Schütze, H. (1999). *Foundations of statistical language processing*. Cambridge: THE MIT PRESS.

McEnery, T., & Wilson, A. (2005). *Corpus Linguistics: an Introduction* (2. Auflage). Edinburgh: Edinburgh University Press.

Meadow, C. T. (1992). *Text Information Retrieval Systems.* San Diego: Academic Press Inc.

Miller, G. A. (1999). Words in WordNet. In Fellbaum, C. (Hrsg.), *WordNet: An electronic lexical database*, (2. Druck) (Seiten 23-46). Cambridge: MIT Press.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11) Seiten 39-41. New York: ACM Press.

Miller, K. J. (1999). Modifieres in WordNet. In Fellbaum, C. (Hrsg.), *WordNet: An electronic lexical database*, (2. Druck) (Seiten 47 - 67). Cambridge: MIT Press.

MIT OCW (2010). Chris Hendrickson: *Project Management for Construction: Organizing the Project Management*. URL - http://pmbook.ce.cmu.edu/index.html

Mitkov, R. (2005). Anaphora Resolution. In Mitkov, R. (Hrsg.), *The Offord Handbook of Computational Linguistics* (Seiten 266-284). Oxford: Oxford University Press.

Moens, M.-F., & De Busser, R. (2006). Information Extraction ans Information Technology. In Moens, M.-F. (Hrsg.), *Information extraction: algorithms and prospects in a retrieval context* (Seiten 1 – 22). Heidelberg: Springer Verlag.

Moens, F. – M., Angheluta, R., & De Busser, R. (2003). Summarization of Texts Found in the World Wide Web. In Abramowicz, W. (Hrsg.), *Knowledge-based information retrieval and filtering from the Web* (Kapitel 5, Seiten 101 -120) (illustrierte Ausgabe). Heidelberg: Springer Verlag.

Moens, F. – M., & Angheluta, R. (2003). Concept extraction from legal cases: the use of a statistic of a coincidence. In *Proceedings of the 9th international conference of Artificial intelligence and law* (Scotland, United Kingdom, 2003), Seiten 142 – 146. New York: ACM Press.

Neumann, G. (2001). Informationsextraktion. In Carstensen, K. U., Anwendungen. In Carstensen, K. U., Ebert, C., Endriss, C., Jekat, S., Klabunde, R., & Langer, H. (Hrsg.), *Computerlinguistik und Sprachtechnologie: eine Einführung* (Seiten 409 - 521). Heidelberg: Spektrum Akademischer Verlag.

Oxford (2010). AskOxford: Language Facts. URL - http://www.askoxford.com/oec/mainpage/oec02/?view=uk, (Zugriffsdatum: 19. Jänner 2010)

Paul, H. (2000). Deutsche Gramatik III. In Ludger Hoffmann (Hrsg.), *Sprachwissenschaft: Ein Reader* (2. Verbesserte Auflage, Seiten 494 - 503). Berlin: Walter de Gruyter.

Pfister, B., & Kaufmann, T. (2008). *Sprachverarbeitung: Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. Heidelberg: Springer Verlag.

Porter, M. F. (1997). An algorithm for suffix stripping. In Sparck, K., & Wilett, P., *Readings in Information Retrieval (Seiten 313 - 316)*. San Francisco: Morgan Kaufmann Publishers.

Ramirez, P. M., & Mattmann, C. A. (2004). ACE: Improving search engines via automatic concept extraction. In *Proceedings of the 2004 IEEE international conference on information reuse and integration*, IRI 2004 (Las Vegas, Nevada, 2004), Seiten 229 - 234.

Roussey, C., Calabretto, S., Harrathi, F., & Gammoudi, M. (2006). Multilingual Indexing Based on Ontologies. In Ghodous, P., Dieng-Kuntz, R., & Geilson, L. (Hrsg), *Leading the Web in concurrent engineering: next generation concurrent engineering*, Seiten 418 - 425. Amsterdam: IOS Press.

Schiehlen, M., & Klabunde, R. (2001). Semantik. In Ebert, C., Endriss, C., Methoden. In Carstensen, K. U., Ebert, C., Endriss, C., Jekat, S., Klabunde, R., & Langer, H. (Hrsg.), *Computerlinguistik und Sprachtechnologie: eine Einführung* (Seiten 246 - 304). Heidelberg: Spektrum Akademischer Verlag.

Sebastiani, F. (2002). Machine Learning in Automatic Text Categorization. *ACM Computing Surveys, 34* (1), Seiten 1 – 47.

SmILE (2010). XtraK4Me – Extraction of Keyphrases for Metadata Creation. URL - http://smile.deri.ie/projects/keyphrase-extraction, (Zugriffsdatum 22 Februar 2010).

Sparck Jones, K., (1999). Automatic Summarizing: factors and directions. In Mani, I., & Maybury, M. T. (Hrsg.), *Advances in Automatic Text Summarization* (Seiten 1 - 13). Cambridge: The MIT Press.

Sun, B., Mitra, P., Zha, H., Giles, C. L., & Yen, J (2007). Topic segmentation with shared topic detection and alignment of multiple documents. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (Amsterdam, The Netherlands), Seiten 199 – 206. New York: ACM Press.

Stock, W. G. (2007). *Information Retrieval: Informationen suchen und finden*. München: Oldenbourg Wissenschaftsverlag.

Stubbs, M. (2002). *Words and phrases: corpus studies of lexical semantics*. Oxford: Blackwell Publishers.

Turtle, H. & Croft, B. W. (1991). Evaluation of an inference network-based retrieval model. *ACM Transactions and Information Systems, 9* (3), Seiten 187-222.

Turmo, J., Ageno, A., & Català N. (2006). Adaptive Information Extraction. *ACM Computing Surveys, 38* (2), Artikel 4.

Turney, P. D. (2000). Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4), Seiten 303-336.

Turney, P. D. (2003) Coherent Keyphrase Extraction via Web Mining. *Proceedings of IJCAI'03,* (Acapulco, Mexico, 2003), Seiten 434–439.

Villalon, J., & Calvo, R. A. (2009). Concept Extraction from student essays, towards Concept Map Mining. In Proceedings of the 9th IEEE International Conference on Advanced Learning Technologies, ICALT 2009 (Riga, Latvia, 2009). Washington: IEEE Computer Society.

Volmert, J. (2005). Sprache und Sprechen: Grundbegriffe und sprachwissenschaftliche Konzepte. In Volmert, J. (Hrsg.), *Grundkurs Sprachwissenschaft* (5. Auflage) (Seiten 9 - 29). Stuttgart: Willhelm Fink Verlag

Walther, M. (2001). Phonologie. In Ebert, C., Endriss, C., Methoden. In Carstensen, K. U., Ebert, C., Endriss, C., Jekat, S., Klabunde, R., & Langer, H. (Hrsg.), *Computerlinguistik und Sprachtechnologie: eine Einführung* (Seiten 136 - 174). Heidelberg: Spektrum Akademischer Verlag.

Wang, J., Peng, H., & Hu J.-S. (2006). Automatic Keyphrases Extraction from Document Using Neural Network. In Yeung, D. S., Liu, Z.-Q., Wang, X.-Z., & Yan H. (Hrsg.), *Advances in Machine Learning and Cybernetics: 4th International Conference, ICMLC 2005* (Guangzhou, China, 2005) (Revised Selected Papers, Seiten 633 – 641). Heidelberg: Springer Verlag.

Whitley, D. L. (1989). The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, Seiten 116-123. San Francisco: Morgan Kaufmann Publishers Inc.

Wikipedia (2010). Onthologie (Informatik). URL - http://de.wikipedia.org/wiki/Ontologie_%28Informatik%29, (Zugriffsdatum: 13. März 2010).

Wilks, Y., & Catizone, R. (1999). Can we Make Information Extraction More Adaptive? In Pazienza, M. T. (Hrsg.), *Information Extraction: Towards Scalable, Adaptable Systems* (Seiten 1-16). Heidelberg: Springer Verlag.

Witten, I. H., Paynter, G., Frank, E., Gutwin, C., and Nevill-Manning, C. G. (1999). Kea: Practical automatic keyphrase extraction. In *Proceedings of the 4th ACM International Conference on Digital Libraries*, Seiten 254-255, Berkeley, C, United States.

WordNet (2010). WordNet: A lexical database for English. URL - http://wordnet.princeton.edu/, (Zugriffsdatum: 19. Februar 2010).

Wu, M., Li, W., Lu, Q., & Wong, K.-F. (2007). Event-Based Summarization Using Time Features. In Gelbukh, A. (Hrsg), *Computational Linguistics and Intelligent Text Processing, Proceedings of 8th International Conference, CICLing 2007* (Mexiko City, Mexiko, 2007), Seiten 563 - 574. Heidelberg: Springer Verlag.

Wurzel, W. U. (2000). Der Gegenstand der Morphologie. In Booij, G. E., Lehmann, C., Mugdan, J. (Hrsg.), *Morphologie: Internationales Handbuch zur Flexion und*

Wortbildung (Morphology: International handbook on inflection and word-formation) (Seiten 1 - 15). Berlin: Walter de Gruyter.

Zhai, X. (2008). Statistical Language Models for Information Retrieval: A Critical Review. *Foundations and Trend® in Information Retrieval, 2* (3), Seiten 137-213.

Zimmermann, H.-J. (2001). *Fuzzy Set Theory: and its applications* (4. Ausgabe). Boston: Kluwer Academic Publishers.

11 Anhang

11.1 Evaluierung

Evaluierungsdokument:

Project Management

The management of construction projects requires knowledge of modern management as well as an understanding of the design and construction process. Construction projects have a specific set of objectives and constraints such as a required time frame for completion. While the relevant technology, institutional arrangements or processes will differ, the management of such projects has much in common with the management of similar types of projects in other specialty or technology domains such as aerospace, pharmaceutical and energy developments.

Generally, project management is distinguished from the general management of corporations by the mission-oriented nature of a project. A project organization will generally be terminated when the mission is accomplished. According to the Project Management Institute, the discipline of project management can be defined as follows:

Project management is the art of directing and coordinating human and material resources throughout the life of a project by using modern management techniques to achieve predetermined objectives of scope, cost, time, quality and participation satisfaction.

By contrast, the general management of business and industrial corporations assumes a broader outlook with greater continuity of operations. Nevertheless, there are sufficient similarities as well as differences between the two so that modern management techniques developed for general management may be adapted for project management.

Supporting disciplines such as computer science and decision science may also play an important role. In fact, modern management practices and various special knowledge domains have absorbed various techniques or tools which were once identified only with the supporting disciplines. For example, computer-based information systems and decision support systems are now common-place tools for general management. Similarly, many operations research techniques such as linear programming and network analysis are now widely used in many knowledge or application domains.

Specifically, project management in construction encompasses a set of objectives which may be accomplished by implementing a series of operations subject to resource constraints. There are potential conflicts between the stated objectives with regard to scope, cost, time and quality, and the constraints imposed on human material and financial resources. These conflicts should be resolved at the onset of a project by making the necessary tradeoffs or creating new alternatives. Subsequently, the functions of project management for construction generally include the following:

- Specification of project objectives and plans including delineation of scope, budgeting, scheduling, setting performance requirements, and selecting project participants.
- Maximization of efficient resource utilization through procurement of labor, materials and equipment according to the prescribed schedule and plan.
- Implementation of various operations through proper coordination and control of planning, design, estimating, contracting and construction in the entire process.
- Development of effective communications and mechanisms for resolving conflicts among the various participants.

The Project Management Institute focuses on nine distinct areas requiring project manager knowledge and attention:

- Project integration management to ensure that the various project elements are effectively coordinated.
- Project scope management to ensure that all the work required is included.
- Project time management to provide an effective project schedule.
- Project cost management to identify needed resources and maintain budget control.
- Project quality management to ensure functional requirements are met.

- Project human resource management to development and effectively employ project personnel.
- Project communications management to ensure effective internal and external communications.
- Project risk management to analyze and mitigate potential risks.
- Project procurement management to obtain necessary resources from external sources.

These nine areas form the basis of the Project Management Institute's certification program for project managers in any industry.

Trends in Modern Management

In recent years, major developments in management reflect the acceptance to various degrees of the following elements: the management process approach, the management science and decision support approach, the behavioral science approach for human resource development, and sustainable competitive advantage. These four approaches complement each other in current practice, and provide a useful groundwork for project management.

The management process approach emphasizes the systematic study of management by identifying management functions in an organization and then examining each in detail. There is general agreement regarding the functions of planning, organizing and controlling. A major tenet is that by analyzing management along functional lines, a framework can be constructed into which all new management activities can be placed. Thus, the manager's job is regarded as coordinating a process of interrelated functions, which are neither totally random nor rigidly predetermined, but are dynamic as the process evolves. Another tenet is that management principles can be derived from an intellectual analysis of management functions. By dividing the manager's job into functional components, principles based upon each function can be extracted. Hence, management functions can be organized into a hierarchical structure designed to improve operational efficiency. The basic management functions are performed by all managers, regardless of enterprise, activity or hierarchical levels. Finally, the development of a management philosophy results in helping the manager to establish relationships between human and material resources. The outcome of following an established philosophy of operation helps the manager win the support of the subordinates in achieving organizational objectives.

The management science and decision support approach contributes to the development of a body of quantitative methods designed to aid managers in making complex decisions related to operations and production. In decision support systems, emphasis is placed on providing managers with relevant information. In management science, a great deal of attention is given to defining objectives and constraints, and to constructing mathematical analysis models in solving complex problems of inventory, materials and production control, among others. A topic of major interest in management science is the maximization of profit, or in the absence of a workable model for the operation of the entire system, the suboptimization of the operations of its components. The optimization or suboptimization is often achieved by the use of operations research techniques, such as linear programming, quadratic programming, graph theory, queuing theory and Monte Carlo simulation. In addition to the increasing use of computers accompanied by the development of sophisticated mathematical models and information systems, management science and decision support systems have played an important role by looking more carefully at problem inputs and relationships and by promoting goal formulation and measurement of performance. Artificial intelligence has also begun to be applied to provide decision support systems for solving ill-structured problems in management.

The behavioral science approach for human resource development is important because management entails getting things done through the actions of people. An effective manager must understand the importance of human factors such as needs, drives, motivation, leadership, personality, behavior, and work groups. Within this context, some place more emphasis on interpersonal behavior which focuses on the individual and his/her motivations as a socio-psychological being; others emphasize more group behavior in recognition of the organized enterprise as a social organism, subject to all the attitudes, habits, pressures and conflicts of the cultural environment of people. The major contributions made by the behavioral scientists to the field of management include: the formulation of concepts and explanations about individual and group behavior in the organization, the empirical testing of these concepts methodically in many different experimental and field settings, and the establishment of actual managerial policies and decisions for operation based on the conceptual and methodical frameworks.

Sustainable competitive advantage stems primarily from good management strategy. As Michael Porter of the Harvard Business School argues:
Strategy is creating fit among a company's activities. The success of a strategy depends on doing many things well - not just a few - and integrating among them. If there is no fit among activites, there is no distinctive strategy and little sustainability.
In this view, successful firms must improve and align the many processes underway to their strategic vision. Strategic positioning in this fashion requires:

- Creating a unique and valuable position.
- Making trade-offs compared to competitors
- Creating a "fit" among a company's activities.

Project managers should be aware of the strategic position of their own organization and the other organizations involved in the project. The project manager faces the difficult task of trying to align the goals and strategies of these various organizations to accomplish the project goals. For example, the owner of an industrial project may define a strategic goal as being first to market with new products. In this case, facilities development must be oriented to fast-track, rapid construction. As another example, a contracting firm may see their strategic advantage in new technologies and emphasize profit opportunities from value engineering.

Strategic Planning and Project Programming

The programming of capital projects is shaped by the strategic plan of an organization, which is influenced by market demands and resources constraints. The programming process associated with planning and feasibility studies sets the priorities and timing for initiating various projects to meet the overall objectives of the organizations. However, once this decision is made to initiate a project, market pressure may dictate early and timely completion of the facility.
Among various types of construction, the influence of market pressure on the timing of initiating a facility is most obvious in industrial construction. Demand for an industrial product may be short-lived, and if a company does not hit the market first, there may not be demand for its product later. With intensive competition for national and international markets, the trend of industrial construction moves toward shorter project life cycles, particularly in technology intensive industries.
In order to gain time, some owners are willing to forego thorough planning and feasibility study so as to proceed on a project with inadequate definition of the project scope. Invariably, subsequent changes in project scope will increase construction costs; however, profits derived from earlier facility operation often justify the increase in construction costs. Generally, if the owner can derive reasonable profits from the operation of a completed facility, the project is considered a success even if construction costs far exceed the estimate based on an inadequate scope definition. This attitude may be attributed in large part to the uncertainties inherent in construction projects. It is difficult to argue that profits might be even higher if construction costs could be reduced without increasing the project duration. However, some projects, notably some nuclear power plants, are clearly unsuccessful and abandoned before completion, and their demise must be attributed at least in part to inadequate planning and poor feasibility studies.
The owner or facility sponsor holds the key to influence the construction costs of a project because any decision made at the beginning stage of a project life cycle has far greater influence than those made at later stages. Moreover, the design and construction decisions will influence the continuing operating costs and, in many cases, the revenues over the facility lifetime. Therefore, an owner should obtain the expertise of professionals to provide adequate planning and feasibility studies. Many owners do not maintain an in-house engineering and construction management capability, and they should consider the establishment of an ongoing relationship with outside consultants in order to respond quickly to requests. Even among those owners who maintain engineering and construction divisions, many treat these divisions as reimbursable, independent organizations. Such an arrangement should not discourage their legitimate use as false economies in reimbursable costs from such divisions can indeed be very costly to the overall organization.
Finally, the initiation and execution of capital projects places demands on the resources of the owner and the professionals and contractors to be engaged by the owner. For very large projects, it may bid up the price of engineering services as well as the costs of materials and equipment and the contract prices of all types. Consequently, such factors should be taken into consideration in determining the timing of a project.

Auswertung der Evaluierung – Tabellen:

Kapitel 1	Person 1	Person 2	Person 3	Person 4	Person 5	durchsch. Relevanz
project objectives and plans	100	80	90	25	100	79%
construction projects	100	90	100	90	100	96%
modern management techniques	100	20	90	10	100	64%
the management	80	75	80	65	70	74%
the design and construction process	90	60	80	80	90	80%
human and material resources	90	75	80	70	85	80%
the mission	20	10	10	5	20	13%
institutional arrangements	50	40	60	30	40	44%
the project management institute	10	0	5	5	10	6%
project scope management	70	50	60	40	80	60%
Kapitel 2						
the managment science and decision support	90	90	95	80	100	91%
project managment	100	80	100	90	100	94%
the management process approach	90	80	95	90	100	91%
management functions	95	50	100	85	80	82%
monto carlo simulation	85	70	80	90	100	85%
management	70	60	75	40	70	63%
actual managerial policies and decisions	70	50	60	40	80	60%
operation and productions	60	20	40	10	50	36%
human and material resources	85	75	90	60	90	80%
the harvard business school	5	0	10	0	10	5%
Kapitel 3						
Capital projects	80	40	50	20	70	52%
construction costs	90	90	90	80	100	90%
engineering and construction costs	100	80	80	60	100	84%
market demands and resource constraints	100	90	50	75	100	83%
construction	40	10	30	5	50	27%
planing and feasability	60	50	80	20	90	60%
earlier facility operation	20	0	30	5	10	13%
some owners	10	10	20	0	30	14%
the stratetic plan	100	85	100	90	100	95%
the facility	20	0	10	0	10	8%
Relevnaz	**69,33**	**51,00**	**64,67**	**45,33**	**71,17**	53%
durchschnittliche Relevanz						**60,3**

Kapitel 1	Übereinstimmungen		
person 1	top 5	top 10	untereinander
Project management	0,5	0,5	1
similarities as well as differences	0	0	0
computer-based information systems and decision support systems	0	0	0
functions of project management	0,5	0,5	1
focuses on nine distinct areas	0	0	0
person 2	**1**	**1**	**2**
Modern management techniques	1	1	1
supporting disciplines	0	0	0
stated objectives	0,5	0,5	0
coordination	0	0	0
resolving conflicts	0	0	0
person 3	**1,5**	**1,5**	**1**
General management	0,5	0,5	0,5
project management	0,5	0,5	1
project objectives	0,5	0,5	0,5
modern management techniques	1	1	1
functions of project management	0	0	1
person 4	**2,5**	**2,5**	**4**
General management of corporations	0	0	0,5
absorbed various tools	0	0	0
maximization of efficient resource utilization	0	0	1
development of effective communications	0	0	1
directing and coordinating resources	0	0	0,5
person 5	**0**	**0**	**3**
art of directing and coordinating human and material resources	0	1	0,5
Specification of project objectives and plans	1	1	0,5
Maximization of efficient resource utilization	0	0	1
Implementation of various operations	0	0	0
Development of effective communications	0	0	1
	1	**2**	**3**
Durchschnitt Übereinstimmung	1,2	1,4	1,6

Kapitel 2	Übereinstimmungen		
person 1	top 5	top 10	untereinander
Acceptance tovarious degrees	0	0	0
hierarchical structure	0	0	0
support systems	0	0	0
optimization or suboptimization	0	0	0
strategic positioning	0	0	1
person 2	**0**	**0**	**1**
Behavioural sience	0	0	0
competitive advantage	0	0	0
interrelated processes	0	0	0
human resource development	0	0,5	0,5
strategic position	0	0	1
person 3	**0**	**0,5**	**1,5**
Modern management	1	1	0
management functions	1	1	0
management science	0,5	0,5	0,5
project manager	0,5	0,5	0
successful firms	0	0	0
person 4	**3**	**3**	**0,5**
Systematic study of management	0	0	0
managers job is regarded as coordinating	0	0	0
development of a management philosophy	0	0	0
maximization of profit	0	0	0
creating a unique and valuable position	0	0	0
person 5	**0**	**0**	
the management process approach	1	1	0
the management science and decision support approach,	1	1	0,5
the behavioral science approach for human resource development	0	0	0,5
strategic position of their own organization	0	0	0,5
trying to align the goals and strategies	0	0	0
	2	**2**	**1,5**
Durchschnitt	1	1,1	0,9

Kapitel 3	Übereinstimmungen		
person 1	top 5	top 10	untereinander
Capital projects	1	1	1
various types of construction	0,5	0,5	0
planning and feasibility	0	1	0,5
project life cycle	0	0	1
resources of the owner	0	0,5	0
person 2	**1,5**	**3**	**2,5**
Capital projects	1	1	
project programming	0	0	1
strategic plan	0	1	0
market	0,5	0,5	0,5
construction	1	1	0,5
person 4	**2,5**	**3,5**	**2**
Hit the market first	0	0	0
shorter project life cycles	0	0	1
facility operation often justify the increase in construction costs	0,5	0,5	0,5
decision at the beginning has greater influence then those later	0	0	0
provide adequate planning and feasability	0	0,5	0,5
person 5	**0,5**	**1**	**2**
strategic plan of an organization	0	0,5	0,5
market demands	0,5	0,5	0
construction costs	1	1	0,5
operating costs	0	0	0
demands on the resources of the owner	0	0	0
	1,5	**2**	**1**
Durchschnitt	1,1	1,5	1,2

Werte für das gesamte Dokument	top 5	top 10	untereinander
Übereinstimmungen Durchschnitt absolut	1,1	1,33	1,57
Übereinstimmungen Durchschnitt prozentuell	22%	27%	31%

11.2 CD