Entwurf und Entwicklung von Konzepten zur automatischen Fragengenerierung

Masterarbeit

an der

Technischen Universität Graz

vorgelegt von

Klaus Lankmayr

April 2010

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

Institut für Informationssysteme und Computer Medien (IICM),
Technische Universität Graz
A-8010 Graz, Austria

© Copyright 2010, Klaus Lankmayr

Design and Development of Concepts for Automatic Exercise Creation

Master's Thesis

at

Graz University of Technology

submitted by

Klaus Lankmayr

April 2010

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

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

© Copyright 2010, Klaus Lankmayr

Kurzfassung

In den letzten Jahren hat sich, aufgrund der Möglichkeiten, die das World Wide Web bietet, die Art des Informationsangebotes und der -nachfrage stark gewandelt. Zusätzlich ändern sich die Anforderungen an die Gesellschaft durch den rasanten technologischen Fortschritt und erfordern dadurch in vielen Bereichen des alltäglichen Lebens ein lebenslanges Lernen. Im Zuge dieser Entwicklung ist die Gesellschaft dazu aufgefordert sich eigenständig weiterzubilden, die bestehenden Ressourcen optimal zu nutzen und das eigenständig erworbene Wissen selbstständig zu evaluieren, da in den meisten Fällen die Überprüfung durch Fachpersonal nahezu unmöglich ist. Von daher ist es erforderlich, Methoden zu entwerfen, die es ermöglichen, eine so weit als möglich automatisierte Wissensüberprüfung durchzuführen.

Um diesen Anforderungen gerecht zu werden bedarf es Methoden, um von beliebigen natürlich sprachlichen Inhalten die wesentlichen Konzepte zu extrahieren und basierend auf diesen, Aufgaben zu erstellen, Assessments durchzuführen und letztendlich den Lernprozess durch geeignetes Feedback iterativ zu unterstützen. Diese Masterarbeit stellt nach Aufarbeitung der theoretischen Grundlagen von Natural Language Processing und Assessment die wichtigsten Forschungsansätze in Bezug auf Konzeptextraktion, automatische Aufgabenerstellung und Assessmentsysteme vor. Die daraus gewonnen Erkenntnisse werden kritisch untersucht und auf deren Zweckdienlichkeit für einen möglichst allgemeinen Lösungsansatz eines Gesamtsystems geprüft.

Die nach diesen Analysen als nützlich klassifizierten Lösungsansätze werden in der Implementierung eines Prototyps, dem Automatic Question Creator, umgesetzt. Das entwickelte Programm ist aufgrund von vorhergehenden statistischen und semantischen Analysen in der Lage, relevante Konzepte zu bestimmen und automatisiert Open Ended, Fill In The Blank, Single Choice und Multiple Choice Aufgaben im QTI Standard zu erstellen, um eine mögliche Integration in andere Systeme zu ermöglichen.

Die Ergebnisse einer kleinen Evaluierung besagen, dass die vom implementierten System generierten Fragen von unabhängigen Testpersonen in allen Bereichen als sehr zufrieden stellend empfunden wurden. Diese Ergebnisse weißten im Vergleich zu den Ergebnissen jener Fragen, die händisch erstellt wurden, nur minimale Unterschiede auf. Im Fokus der Untersuchung standen dabei die Relevanz zum Thema, der Schwierigkeitsgrad, das ausgewählte Konzept, die Referenzantwort und die Distraktorenqualität.

Abstract

In recent years, due to the opportunities provided by the World Wide Web, the kind of information supply and demand has rapidly changed. In addition, the demands on society change by the dynamic technological progress and thus require life long learning in many areas of everyday life. In the course of this advancement society is encouraged to continue its studies independently, to utilize the existing resources optimally and to evaluate the acquired knowledge itself because in most cases, the review by qualified personnel is almost impossible. Therefore it is necessary to design methods which allow an automated verification of knowledge as far as possible.

To meet these requirements it needs methods to extract the main concepts from any natural language contents, to create tasks based on these concepts, carry out assessments and eventually to support the process of learning by appropriate feedback iteratively. This master's thesis provides, towards reconditioning of the theoretical fundamentals of natural language processing and assessment, the main approaches related to concept extraction, automatic task creation and assessment systems. The gained insights are critically examined and tested for their suitability for a preferably general solution approach of an overall system.

The solutions that are classified as useful by these analyses are assembled in the implementation of a prototype; the Automatic Question Creator. The developed program is, based on previous statistic and semantic analysis, able to determine the relevant concepts and to construct open-ended, fill-in-the-blank, single-choice and multiple-choice tasks in the QTI standard, to allow a possible integration into other systems.

The results of a small evaluation show, that questions which are generated by the implemented system are classified as satisfactory from independent testees in all areas. These results reveal only minimal differences in comparison to the results of those questions that were created by hand. This investigation focused on the relevance to the topic, the degree of difficulty, the selected concept, the reference response and the distractor quality.

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 besonderer Dank gilt meinen Eltern, Anna und Ernst, die es mir überhaupt ermöglicht haben zu studieren und mich stets in jeglicher Art und Weise unterstützten. Außerdem möchte ich meiner Freundin Silvia, meinem Sohn Dennis, meinen Brüdern Markus und Rainer ganz herzlich für die aufmunternden Stunden in schwierigen Zeiten meinen Dank aussprechen.

Darüber hinaus möchte ich sehr herzlich meinem, mich beinahe das ganze Studium begleitenden, Studienkollegen Joachim Weinhofer danken, der parallel zu mir seine Masterarbeit über die Extraktion semantischer Daten geschrieben hat, gemeinsam mit mir and den zentralen Elementen des Automatic Question Creator gearbeitet hat und dessen Ergebnisse aus Implementierungssicht ich für die Umsetzung meines Teils der praktischen Arbeit Nutzen konnte.

Letztendlich möchte ich auch von ganzem Herzen meinem Betreuer Dipl.-Ing. Dr.techn. Christian Gütl danken, der mir einerseits die Möglichkeit gegeben hat diese spannende Arbeit zu verfassen und mich andererseits immer so gut als möglich betreut und motiviert hat. Darüber hinaus möchte ich noch Herrn Ass.-Prof. Mag. Dr.phil. Rudolf Muhr in dieser Danksagung erwähnen, der hilfreiche Tipps zur Umsetzung der Theorie geben konnte.

Lankmayr Klaus
Graz, Österreich, April 2010

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 *Einleitung*

Im Zuge dieser Arbeit sollen Konzepte entworfen werden, die es ermöglichen, aus natürlich sprachlichen Texten beliebiger Kategorien voll automatisiert die wesentlichen Inhalte zu erfassen und in weiterer Folge aus jenen Inhalten Fragestellungen beziehungsweise Aufgaben verschiedenster Art zu generieren. Bevor näher auf den Aufbau und die Struktur dieser Arbeit eingegangen wird, wird kurz aufgezeigt, aus welchen Gründen diese Problem- und Aufgabenstellung sinnvoll und nötig ist.

1.1 Motivation und aktuelle Situation

In Anbetracht der schnelllebigen Zeit und der Voraussetzung von lebenslangem Lernen ist es im Zeitalter der digitalen Informationsvielfalt, in welchem das Informationsangebot, unter anderem bedingt durch den beinahe exponentiellen Anstieg von Web Sites und laut Hausser (2000) auch der zweiten Gutenbergeschen Revolution, rasend schnell wächst, sehr schwierig, sich die wesentlichen Informationen selbst anzueignen. Die Autoren Berlak et al. (1992) sind ebenfalls der Ansicht, dass es aufgrund der Komplexität der Informationsvielfalt eine große Herausforderung ist, relevante Inhalte aufzufinden, wobei einerseits der Umstand, dass viele Inhalte dieses Informationsangebotes verfälscht und nicht auf Richtigkeit verifiziert sind, das Lernen erschweren, andererseits den Nutzern jener interaktiven Informationsquellen die Möglichkeit fehlt, das sich selbst angeeignete Wissen zuverlässig zu überprüfen.

Die Betroffenen dieses Lernzyklus sind neben den Lernenden auch die Lehrenden, welche ihren Unterricht, neben den traditionellen Lehrmethoden, aufgrund der wachsenden Zahl von Informationssuchenden und dem Wandel der Informationsbeschaffung, zunehmend auf e-Learning Einheiten verschieben müssen. Da das Assessment der gelehrten Inhalte vor allem in Bezug auf die Aufgabenerstellung und die anschließende Auswertung jener Aufgaben sehr viel Zeit in Anspruch nimmt, besteht ein großes Bedürfnis nach unterstützenden Methoden und Hilfsmitteln. Auch hierbei würde eine vollständig automatisierte Aufgabenerstellung und in weiterer Folge eine computergestützte Auswertung, sowie ein vom System selbstständig generiertes Feedback, einen unschätzbaren Vorteil bringen. Untermauert wird diese Aussage von Mason und Grove-Stephenson (2002), die verlautbaren, dass Lehrer in England etwa 30 Prozent der Arbeitszeit für die Beurteilung aufwenden.

1.2 Struktur der Arbeit

In dieser Masterarbeit werden die theoretischen Hintergründe von Natural Language Processing und Assessment erklärt, der Forschungsstand dahingehend aufgezeigt, daraus einige Konzepte entwickelt sowie eine Implementierung vorgestellt.

Das Kapitel 2 dieser Arbeit beschäftigt sich mit dem Thema Natural Language Processing, wobei wichtige Grundbegriffe definiert und Verarbeitungsschritte aufgezeigt werden. Zusätzlich werden stellvertretend einige, zu der Aufgabenstellung analoge, Anwendungsbeispiele gegeben, bevor letztendlich auf Assessment im Bereich von E-Learning übergegangen wird. Abschließend wird in diesem Abschnitt kurz dargelegt, welche Arten von e-Assessment in welcher Art und Weise zum Einsatz kommen und welche Anforderungen ein Assessment System erfüllen soll.

Das Kapitel 3 umfasst die Vorstellung aktueller Anwendungen und des Forschungsstandes in den drei Bereichen automatische Termextraktion, automatische Aufgabenerstellung und automatisches Assessment. Die einzelnen Ansätze werden im Hinblick auf deren Brauchbarkeit untersucht, wobei die Vor- und Nachteile aufgezeigt werden.

In Kapitel 4 sollen die Anforderungen an ein System, das die bis dahin gewonnen Erkenntnisse sinnvoll einsetzt, kombiniert und verbessert, beschrieben werden. Zusätzlich wird ein mögliches konzeptionelles Design, welches einen so weit als möglich optimalen Lösungsweg für die Aufgabenstellung darstellt, erarbeitet.

Kapitel 5 stellt die in der begleitenden Implementierung verwendeten Tools und Frameworks vor und es werden die darin enthaltenen Algorithmen sowie Plugins näher vorgestellt.

In Kapitel 6 wird die abgelieferte Implementierung in Bezug auf das konzeptionelle Design, Implementierungsdetails, Probleme der Implementierung und die Sichtweise des Nutzers vorgestellt. Darüber hinaus werden offene Erweiterungsmöglichkeiten beschrieben und eine kleine Evaluierung durchgeführt.

Kapitel 7 zeigt die gewonnenen Erkenntnisse des Verfassers dieser Arbeit in den fachspezifischen Gebieten des Natural Language Processing und Assessment, aber auch in Bezug auf das Recherchieren und Verfassen einer wissenschaftlichen Arbeit auf.

Im Kapitel 8 werden die wesentlichen Ideen und Konzepte dieser Arbeit wiederholt und ein Ausblick darauf gegeben, auf welche Kernbereiche sich die Forschung in Zukunft fokussieren wird und worin Verbesserungspotential besteht.

2 *Natural Language Processing und Assessment*

Bevor näher auf die Aufgaben- und Problemstellung der Entwicklung von Konzepten zur automatischen Fragengenerierung eingegangen werden kann, ist es erforderlich einige richtungweisende Ansätze zum Thema Natural Language Processing und Assessment herauszuarbeiten. Zunächst werden die sprachlichen Grundlagen, deren Verarbeitung, sowie Assessment im Allgemeinen abgehandelt, um aus den gewonnenen Erkenntnissen Querverbindungen zur Aufgabenstellung herstellen zu können.

2.1 Natural Language Processing

In diesem Abschnitt werden zu Beginn, zu Gunsten des Verständnisses der Thematik, wichtige Begriffe und Definitionen zum Thema Natural Language Processing dargelegt. Danach wird beschrieben inwieweit die Vorverarbeitung von Texten als eine Grundvoraussetzung für weiterführende Analysen jener Texte von Nöten ist und in welchem Ausmaß diese zu erfolgen haben. Anschließend wird näher auf die grundlegenden Konzepte der statistischen Sprachverarbeitung eingegangen, welche eine Voraussetzung für weitere Überlegungen, vor allem in Bezug auf Informationsextraktion, darstellen. Nachfolgend werden stellvertretend einige wichtige, der Aufgabenstellung ähnliche, Ansätze in Bezug auf die Sprach- beziehungsweise Textverarbeitung und letztendlich gängige Methoden zur Gewinnung von Information aus natürlich sprachlichen Texten präsentiert.

2.1.1 Allgemeines

Um die prinzipielle Vorgangsweise beim Natural Language Processing zu erläutern ist es hilfreich, sich die, analog zur maschinellen Textverarbeitung beziehungsweise dem maschinellen Textverständnis, durchzuführenden Methoden und Prozesse beim Menschen anzusehen. Die physischen Rohdaten werden auf verschiedenen Ebenen im menschlichen Gehirn aufgenommen, strukturiert, weiterverarbeitet und letztendlich ausgewertet.

Diese Prozesse werden beim Menschen vollkommen adaptiv gesteuert und die dabei gewonnenen Informationen in Wechselwirkung mit dem kognitiven Gedächtnis ausgewertet. Dabei spielen in Bezug auf die kognitiven Prozesse vor allem bereits Erlerntes, aber auch auf das Gedächtnis basierende, komplexe weiterführende Vorstellungen und Ideen eine große Rolle. Derartige Verarbeitung von vorliegenden Rohdaten sind mehr oder weniger äquivalent zu den Schritten, welche nachfolgend aufgezählt werden (vgl. Haton, 1987).

2.1.2 Begriffserklärung und Definitionen

In diesem Unterkapitel werden einige für das weitere Verständnis des Natural Language Processing unabdingbare Begriffe erklärt und definiert. Die vorgestellten Begriffe bedienen sich vornehmlich regelbasierter Methoden, die einzige Ausnahme stellt laut Maas (2006) die Phonologie dar, welche mit autosegmentalen Ansätzen arbeitet. Bevor allgemeine Schemata für Sätze und Texte erarbeitet werden können, ist es nötig, sich mit den einzelnen Wörtern eines Textes zu beschäftigen, welche durch die Morphologie und die Syntax beschrieben werden.

2.1.2.1 Morphologie

Der Autor Karatas (2005) liefert zu dem Begriff Morphologie in der Computerlinguistik eine sehr treffende Definition, nämlich dass jene die Erscheinungsformen, die Struktur und die Bauformen von Wörtern beschreibt. Dies geschieht mitunter durch das Zusammensetzen mehrerer Morpheme, welche die kleinsten bedeutungstragenden Einheiten einer Sprache sind und nicht weiter zerlegt werden können, ohne deren Aussagekraft zu verlieren (vgl. Augst, 1975). Das heißt, dass ein Wort iterativ sukzessive segmentiert werden kann, bis keine weitere Reduktion möglich ist, ohne Information einzubüßen.

Weiterführend schreibt Karatas (2005), dass ein Morphem sowohl eine lexikalische als auch eine funktionale Bedeutung hat, wobei erstere als eine inhaltliche und zweitere als grammatikalische Bedeutung anzusehen sind, welche nachfolgend genauer beschrieben werden.

2.1.2.2 Lexika

Beim Begriff Lexikon werden grundsätzlich zwei Teilbereiche, die Lexikographie und die Lexikologie, unterschieden, welche der Autor Hausser (2000) wie folgt näher beschreibt: *„Die Lexikographie beschäftigt sich mit den Prinzipien der lexikographischen Kodierung und der Struktur lexikalischer Einträge und ist ein praktisch orientiertes Randgebiet der Sprachwissenschaften.“* (S. 74) Die Lexikologie hingegen „*untersucht den Wortschatz einer Sprache in Hinblick auf ihre interne Bedeutungsstruktur (...)*.“ (S. 74) Für eine automatische Textanalyse sind elektronische Wörterbücher unabdingbar, Hausser benennt dies *mining of dictionaries* und führt als Beispiel das Oxford English Dictionary an (vgl. Hausser, 2000).

2.1.2.3 Syntax

Die Syntax beschreibt laut dem Autor Lyons (1995) die Theorie des Zusammenfügens von Wörtern, wobei den Morphemen demnach eine Funktion, etwa jene des Subjektives, gegeben wird. Jedem Wort wird dabei eine Distributionsklasse zuge-

wiesen, welche die anzuwendenden grammatikalischen Regeln vorgibt. Jene konkreten Bedeutungen ergeben in Summe den tatsächlichen Inhalt eines Satzes, wobei die syntaktischen Einheiten oftmals mehrdeutig sein und die daraus abgeleiteten Inhalte stark subjektiv interpretiert werden können (vgl. Pospiech, 2005).

2.1.2.4 Semantik

Unter dem Begriff der Semantik verstehen die Autoren Klabunde und Schiehlen (2001) einerseits die Bedeutung von Wörtern, andererseits aber auch jene von Sätzen und ganzen Texten. Weiters wird erwähnt, dass die Semantik ohne Interpretation nicht möglich ist und dass diese wiederum stark von Logik geprägt wird. Um einzelne Sätze oder auch Passagen in Bezug auf die vermittelte Semantik zu verstehen, ist es oftmals von Nöten, bestimmte situative Parameter vorübergehend auszublenden und die Gesamtheit zu betrachten (vgl. Klabunde und Schiehlen, 2001). Ähnlich dem Vorgang des Informationserfassens und -verarbeitens beim Menschen, wie kurz in Kapitel 2.1.1 erläutert, ist die Wahrnehmung semantischer Informationen von subjektiven Eigenschaften wie Logik und bereits erworbenen Wissen abhängig.

2.1.2.5 Pragmatik

Die Pragmatik bedient sich ähnlicher Werkzeuge wie die Semantik und befasst sich, den Autoren Klabunde und Schiehlen (2001) zufolge, ebenso mit der Bedeutung natürlicher Sprache, wobei der Fokus auf dem sprachlichen Handeln liegt. Hierbei nimmt die Grammatikalisierung von bestimmten Teilaspekten des Kontextes eine wichtige Rolle ein und Konzepte wie Implikaturen, Indikatoren sowie Deixis sind Voraussetzungen für das pragmatische Verständnis (vgl. Vater, 2002).

Der Autor Hausser (2000) schreibt darüber hinaus dass die Pragmatik eine Interaktion zwischen Kontext und Ausdruck darstellt und sich damit beschäftigt, welchen Einfluss Wörter und Phrasen in einem spezifischen Kontext auf den Inhalt haben.

2.1.2.6 Phonologie

Da das Gesamtkonzept der Phonologie zur Gruppe der wichtigsten Ansätze der Sprachtechnologie gehört, wird dieser Begriff nachfolgend kurz definiert, obwohl laut dem Autor Maas (2006) der Bezug zur Computerlinguistik fehlt und demnach die Phonologie für die Aufgabenstellung dieser Arbeit keinen Nutzen bringen kann.

Die Phonologie beschäftigt sich, dem Autor Hall (2000) nach, mit der Lehre von Sprachlauten, wobei vor allem zwei wesentliche Aspekte von großer Bedeutung sind. Einerseits ist das der physische, also der artikulatorische, auditive und akusti-

sche Aspekt eines Lautes, andererseits wird bei der Phonologie die Systematik einer Sprache betrachtet.

2.1.2.7 Computerlinguistik

Die Computerlinguistik ist ein Teilbereich der Kognitionswissenschaft (cognitve science) und beschäftigt sich laut Hausser (2000) einerseits mit der Sprachproduktion und andererseits mit dem Sprachverstehen. Hausser schreibt weiters, dass sich dieses interdisziplinäre Fach mit der theoretischen Linguistik, der Lexikographie, der Psycholinguistik, Logiken und Analysen der Philosophie, der Textverarbeitung, als auch mit Datenbanken und gesprochener Sprache befasst. Hierbei ist ersichtlich, welche komplexen Zusammenhänge in der Computerlinguistik auszuwerten sind und welche Rechenleistungen einem Computer diesbezüglich bei der Datenverarbeitung abverlangt werden.

Abschließend sei noch erwähnt dass die aufgeführten Begriffe keineswegs vollständig sind, sondern im Grunde genommen eine oberflächliche, aber vorübergehend ausreichende Betrachtung darstellen. Um die Liste an Begriffen zu komplettieren müssten alleine in Bezug auf einzelne Wörter noch Erklärungen zu Flexion, Orthographie, etc. definiert werden.

2.1.3 Textuelle Vorverarbeitung und semantische Analysen

Die textuelle Vorverarbeitung ist eine Grundvoraussetzung für die in den folgenden Abschnitten dargelegten NLP Methoden, da die Rohdaten eines Textes ohne etwaige Vorverarbeitung in weiterer Folge sehr schwer zu auszuwerten wären (vgl. Grefenstette und Tapanainen, 1994). Aus diesem Grund ist es für alle Programme, die auf NLP Methoden basieren, hilfreich, den zu analysierenden Text in spezielle Formate zu konvertieren, die es in weiterer Folge ermöglichen, geeignete Metainformationen zu speichern, da jene essentiell für darauf aufbauende Methoden sind. Mit dem Hinzufügen derartiger Informationen konstatiert sich der der Begriff der Annotierung, die laut dem Autor Wilcock (2009) die Aufgabe von syntaktischem und morphologischem Parsing hat.

2.1.3.1 Tokenization

Unter dem Tokenizing versteht man einen Vorgang, der den Text, der computerintern als einzelner String repräsentiert wird, in einzelne Segmente aufspaltet und der die Eigenschaft hat, beim Natural Language Processing die einzige Komponente zu sein, welche immer und zu Beginn ausgeführt werden muss (vgl. Klatt und Bohnet, 2005). Als Segment wird hierbei ein Token, also eine spezielle Zeichenkette, be-

zeichnet, welche ihrerseits ein Wort, Interpunktionszeichen, Abkürzungen und dergleichen darstellt.

Tokenizing wird also in erster Linie dazu genutzt, die einzelnen Wörter zu ermitteln, und hat laut Wilcock (2009) jedoch in weiterer Folge auch die Aufgabe, Satzgrenzen zu identifizieren. Dies erweist sich dahingehend als äußerst schwierige Aufgabe, da etwa Punkte Teil von Abkürzung oder Datumsangaben sein können und häufig die Ursache für falsch detektierte Satzgrenzen oder Mehrdeutigkeiten darstellen. Jene Fehlerquellen können eingegrenzt werden, indem man wie Grefenstette und Tapanainen (1994) zusätzlich Lexika mit gängigen Abkürzungen verwendet.

Im Gegensatz zu der Aussage von Klatt und Bohnet (2005), dass Tokenizing immer zu Beginn durchgeführt werden muss, besteht laut Wilcock (2009) dennoch die Möglichkeit vor dem Tokenizing die Satzgrenzen zu ermitteln. Dieses Verfahren wird beispielsweise in den OpenNLP Tools eingesetzt. Das Ergebnis des Tokenizing dient als Grundlage für den nächsten auszuführenden Schritt, das Part-of-Speech Tagging, wohingegen die Satzgrenzen für das syntaktische Parsing benötigt werden (vgl. Wilcock, 2009).

2.1.3.2 Stoppwörter

Der Autor Baeza-Yates (2004) beschreibt Stoppwörter als Wörter, die keinen bis sehr wenig Inhalt vermitteln oder durch deren Gebrauch sich keine Teilmengen von Dokumenten unterscheiden lassen. Zumeist sind dies funktionelle Wörter oder Akronyme, also Kurzwörter, die aus den Anfangsbuchstaben mehrerer Wörter bestehen (vgl. Baeza-Yates, 2004). Funktionelle Wörter sind in diesem Zusammenhang meist Artikel, Präpositionen und Konjunktionen. Der Sinn hinter dem Konzept der Stoppwörter ist deren Elimination in den weiteren Textanalysen und Betrachtungen, was eine Reduktion der Indexstruktur um bis zu 40 Prozent und dadurch in weiterer Folge einen erheblichen Performancegewinn bewirken kann (vgl. Baeza-Yates und Ribeiro-Neto, 1999).

2.1.3.3 Stemming

Unter Stemming versteht man das meist regelbasierte Rückführen eines Tokens auf seine Grundform, wobei wortinterne Informationen keine Rolle spielen und das Vorgehen je nach Sprache sehr verschieden sein kann. Probleme ergeben sich laut dem Autor Baeza-Yates (2004) aufgrund des fehlenden Kontextes beim Stemmen dann, wenn Präfixe gebildet oder fälschlicherweise Wörter verschmelzt werden, was oftmals darauf zurückzuführen ist, dass mehrdeutige Expressionen möglich sind.

Einer der gebräuchlichsten iterativen, performanten und sehr zuverlässigen Stemmer ist der Porter Stemmer. Dabei stehen in jedem der fünf Iterationsschritte mögliche Vorgehensweisen zur Verfügung und das Ergebnis eines Schrittes wird in der nächsten Entscheidungsstufe weiterverarbeitet, bis keine weiter Anwendung von Regeln mehr möglich ist. Der Porter Stemmer arbeitet mit Suffixlisten, anhand derer verschiedene Endungen in den unterschiedlichen Iterationsstufen eliminiert oder ersetzt werden (vgl. Baeza-Yates und Ribeiro-Neto, 1999; Kowalski, 1997).

Aufgrund der möglichen fehlerhaften Zusammenführung von Wörtern gleichen Wortstamms und divergierender Ergebnisse bei rein regelbasierten Stemmern sind modernere, korpus-basierte Ansätze wie ihn der Autor Stock (2007) beschreibt im Aufwind. Bei diesen Stemming Verfahren werden sprachliche Charakteristika verschiedener Themenschwerpunkte miteinbezogen und dementsprechend einige Schwachstellen von herkömmlichen, regelbasierten Ansätzen beseitigt.

2.1.3.4 Part-of-Speech Tagging

Beim Part-of-Speech Tagging geht es darum, jedem Wort, unter Berücksichtigung des Kontexts, eine Wortklasse zuzuordnen. Laut Giménez und Márquez (2004) ist das POS-Tagging ein fundamentales Problem, da es die Grundlage aller komplexeren Analysen bei größeren Datenmengen ist. Für dieses Problem existieren mannigfaltige Lösungsansätze, wobei die wichtigsten laut den genannten Autoren Hidden Markov Modelle, die Maximale Entropie, transformations-basiertes Lernen, Entscheidungsbäume, AdaBoost, Gedächtnis basiertes Lernen und Support Vector Maschinen sind (vgl. Giménez und Márquez, 2004).

Die Erfolgsrate vom POS-Tagging hängt dabei von der Granularität der verwendeten Tags ab, welche üblicherweise zwischen 50 und 250 Unterscheidungen umfasst, und liegt in der Regel bei etwa 95 Prozent, wohingegen jene bei kritischen Wortklassen, wie etwa Teilen von Adverbialphrasen, bei nur 70 Prozent liegt (vgl. Granger, 2002).

2.1.3.5 Eigennamenerkennung

Die Eigennamenerkennung (Named Entity Recognition) hat den Sinn und Zweck einem Token etwa Namen, Städten, Organisationen oder Produkten zuzuordnen, wobei dieser Vorgang sowohl regelbasiert als auch statistisch durchführbar ist. Ein sehr zuverlässiger regelbasierter, von den Autoren Jackson und Moulinier (2007) genannter, Algorithmus wendet im ersten Schritt heuristische Regeln auf den Kontext eines Wortes an, wobei unterstützend Listen von bereits klassifizierten Wörtern verwendet werden. Ist dies für alle potentiellen Eigennamen durchgeführt worden, so werden all jene Wörter im Text erneut klassifiziert, indem trainierte statistische

Modelle, die bis zu diesem Zeitpunkt vorgeschlagenen Wörter, auswerten. Im drauffolgenden Schritt werden alle möglichen, noch nicht erkannten, Eigennamen nur anhand der vorgefertigten Listen ohne kontextuelle Einschränkungen annotiert und schlussendlich noch Wörter in den Überschriften mit Einträgen im Text verglichen und bei Bedarf gekennzeichnet (vgl. Jackson und Moulinier, 2007).

2.1.3.6 Chunk Parsing

Klabunde und Schiehlen (2001) schreiben über das Chunk Parsing, auch partielles oder shallow Parsing genannt, dass man hierbei untergeordnete Teilstrukturen identifiziert und phrasalen Tags zuordnet. Das Ziel hinter dieser Überlegung ist es, die nicht performanten und nicht immer zielführenden vollständigen Parser abzulösen und nur lokale syntaktische Abhängigkeiten abzuhandeln. Die sich daraus ergebenden Einsatzgebiete sind meistens das Information Retrieval und die automatische Textzusammenfassung. Laut Klabunde und Schiehlen (2001) kann ein auf Chunks geparstes Dokument durchaus als Grundlage für einen vollständigen Parser dienen. Neben dem Vorteil, unter bestimmten Voraussetzungen, im Vergleich zum vollständigen Parsen, bessere Ergebnisse bei weniger Ressourcenverbrauch zu liefern, ist dieses Verfahren konsistent in Bezug auf Satz- und Rechtschreibfehler (vgl. Hess und Clematide, 2008).

Hess und Clematide (2008) definieren Chunks als eine nicht rekursive, nicht exhaustive und nicht überlappende Region eines Textes, welche sich durch reguläre Ausdrücke beschreiben lassen. Um diese zu ermitteln, werden im ersten Schritt nur Nominalchunks ausgewertet und in weiterer Folge mit Hilfe der Wortumgebung erweitert. Probleme ergeben sich bei dieser Methode dann, wenn einerseits ein Muster auf verschieden lange Teile eines Textes anwendbar ist oder andererseits jeweils ein Muster beider Schritte auf eine Phrase zutrifft (vgl. Hess und Clematide, 2008).

2.1.3.7 Koreferenzauflösung

Die Koreferenzauflösung hat das Ziel, mehrere Phrasen oder Wörter, ein und dem selben von den Autoren Williams und Poulovassilis (2008) genannten real-word entity zuzuordnen. Über den gesamten Text sollen demnach Duplikate erfasst und in weiterer Folge die pronominalen Koreferenzen, die Koreferenzen auf Namen und die demonstrativen Koreferenzen durch die referenzierten Wörter austauschbar gemacht werden (vgl. Williams und Poulovassilis, 2008). Das hat den Sinn den Text für semantische Analysen besser vorzubereiten und andererseits auch vor allem in Hinblick auf die, später in dieser Arbeit vorgestellten, Wort- und Satzgewichte bessere Ergebnisse liefern zu können.

Die grundlegende Vorgehensweise zum Auffinden von Anaphern, also von andere sprachliche Einheiten referenzierenden Wörtern, geschieht meist mittels formalen Regeln, die auf morphologischen, semantischen und syntaktischen Analysen beruhen (vgl. Munoz, Saiz-Noeda und Montoyo, 2002). Weiters gibt es auch verstärkt auf Distanzmaße basierende Ansätze wie sie etwa von McCallum und Wellner (2003) beschrieben werden.

2.1.3.8 Bildung von Ontologien

Unter einer Ontologie versteht man Regeln zur Darstellung von Beziehungen in Form von Konzepten für jeglichen semantischen Inhalt. Der Autor Gruber (1993) definiert eine Ontologie als ein eine explizite Spezifikation einer Konzeptualisierung, welche eine systematische Darstellung der Existenz ist[1]. Die sich daraus ergebende Struktur und die sich darin befindenden Relationen werden durch das Vokabular implizit ausgedrückt, woraus sich ergibt, dass sich Ontologien durch die Verwendung von repräsentativen Termen bilden lassen. Bestimmte Definitionen in einem Text verknüpfen demnach Wörter mit einer Struktur, wobei gewisse formale Axiome deren mögliche Interpretation einschränken. Beim Entwurf einer Ontologie ist zu Beginn die grundsätzliche Struktur jenes festzulegen und nach Möglichkeit beim Einfügen einer neuen Instanz ein bestehendes Konzept abzuleiten (vgl. Baader, Horrocks und Sattler, 2004).

Die gängigsten Ontologien für elektronische Inhalte im Web sind SHIQ, RDF und die daraus abgeleitete Web Ontology Language OWL. Dabei wird jedes Objekt einer Klasse zugeordnet und es erhält Eigenschaften von dieser. Sinnvollerweise sind Ontologien in einem XML kompatiblen Format gespeichert und somit leicht erweiterbar und austauschbar (vgl. Horrocks, Patel-Schneider und van Harmelen, 2003).

Der Vorteil beim Einsatz von Ontologien ist, dass der Inhalt unabhängig von den auferlegten Formalismen ist und dass jene eine Konsistenz aber keine Vollständigkeit in Bezug auf das spezifizierte Vokabular und damit verbundenen Aussagen oder Fragen garantieren (vgl. Gruber, 1993).

2.1.4 Statistische Sprachverarbeitung

Die statistische Sprachverarbeitung beschäftigt sich unter Zuhilfenahme von den nachfolgend beschriebenen statistischen Methoden damit, Worthäufigkeiten sowie

[1] *"An ontology is an explicit specification of a conceptualization […] an ontology is a systematic account of Existence.*" (Gruber, 1993, S. 199)

jegliche statistische Regelmäßigkeiten im Text zu erkennen und über diese Eigenschaften die relevanten Inhalte zu bestimmen. Dabei sind weder spezifische Kenntnisse über die Textkategorie noch semantisches, morphologisches oder syntaktisches Wissen Voraussetzung für erfolgreiche Ergebnisse. Wie in dieser Arbeit später noch herausgearbeitet wird, liefert eine Kombination von semantischen, syntaktischen, morphologischen und statistischen Analysen die besten Resultate.

2.1.4.1 Worthäufigkeiten

Worthäufigkeiten können vor allem dazu dienen, relevante Inhalte zu ermitteln, da der Mensch beim Verfassen von Texten, gemäß dem Zipfschen Gesetz, das Prinzip des geringsten Kraftaufwandes verfolgt und die Verwendungshäufigkeit einzelner Wörter nach bestimmten Gesetzmäßigkeiten mit der Textlänge verbunden ist (vgl. Mehler, 2005).

Manning und Schütze (2003) zeigen auf, dass die Anzahl der so genannten *function words*, in Bezug auf Information nicht aussagekräftige Wörter, bedingt durch die sprachliche Konstruktion eines Textes, die Anzahl der *exception words*, also die wirklich informationstragenden Wörter, bei weitem übertrifft. Daraus ist sofort ersichtlich, wie schwer es tatsächlich ist, seltenen Wörtern durch den alleinigen Einsatz von statistischen Methoden eine höhere Relevanz zuzuordnen.

Das Konzept der Worthäufigkeiten lässt sich verbessern, indem anstatt der alleinigen Auswertung der Häufigkeit der gestemmten Wörter auch Wörterbücher und Thesauri verwendet werden (vgl. Meadow, 1992). Man weist diesem Ansatz folgend allen Wörtern Synonyme zu was allein bedingt durch die Variation von Wörtern, die ein Autor verwendet um Eintönigkeiten zu vermeiden, sofort erhebliche Verbesserungen bringt, wenn diese bei der Häufigkeitsanalyse miteinbezogen werden. Darüber hinaus finden sich in Wörterbüchern Definitionen und Verwendungsbeispiele von Wörtern, die bei derartigen Häufigkeitsanalysen ebenso mit ausgewertet werden können und vor allem dann einen Nutzen bringen, wenn die Textkategorie bekannt ist, da sich in diesem Fall das Gewicht einzelner Wörter bei Übereinstimmung mit der Definition eines Wortes mit der gegebenen Kategorie adaptieren lässt.

Ein weit verbreitetes Schema zur Worthäufigkeit ist die tf-idf (term frequency – inverted document frequency), die neben der reinen Worthäufigkeit in einem Dokument oder Text auch die invertierte Dokumentenhäufigkeit berücksichtigt. Die tf-idf berechnet sich laut Baeza-Yates und Ribeiro-Neto (1999) aus dem Produkt der normierten Termhäufigkeit im aktuellen Dokument und der normierten Termhäufigkeit in der gesamten Dokumentenmenge. In Abbildung 1 ist die genaue Berechnung der tf-idf dargestellt, wobei $f_{i,j}$ die normierte Termhäufigkeit ist, N die Anzahl der Dokumente angibt und n_i die Anzahl jener Dokumente abbildet, in welchen der Term vorkommt.

$$w_{i,j} = f_{i,j} \times \log \frac{N}{n_i}$$

Abbildung 1: Berechnung tf-idf (vgl. Baeza-Yates und Ribeiro-Neto, 1999)

Im Laufe dieser Arbeit wird noch das Konzept der Wort- und Satzgewichtung vorgestellt, das mehr oder weniger die Worthäufigkeiten als grundlegenden Ansatz verwendet und in Kombination mit anderen Methoden trotz obig genannter Problematik sinnvolle Ergebnisse liefert.

2.1.4.2 N-Gramme

N-Gramme sind eine Datenstruktur, die anstatt von ganzen Wörtern jeweils einzelne, vom Autor Kowalski (1997) *interword symbols*, also Buchstabensequenzen, genannte Einträge hält, die eine fixe Überlappungslänge besitzen. Die Länge der N-Gramme, die laut einer Studie von Fatah Comlekoglu die besten Resultate liefert, beträgt drei, wobei man jene dann Trigramme nennt.

Im Grunde genommen werden N-Gramme dazu genutzt, die Häufigkeit von Sequenzen zu ermitteln und Suchterme zu erweitern, das Konzept ist aber auch dahingehend einsetzbar, als dass man jene bei der Evaluierung von Antworten im Vergleich mit Referenzantworten einsetzen könnte, um Rechtschreibfehler zu ignorieren, die ansonsten die Ergebnisse negativ beeinflussen würden.

2.1.4.3 Vector Space Model

Beim Vector Space Model werden für verschiedene Dokumente jeweils Vektoren mit Indexwörtern und nicht binären Gewichten aufgestellt und dazu genutzt um ein Ähnlichkeitsmaß dieser Dokumente zueinander zu ermitteln. Unter Indexwörtern versteht man hierbei gestemmte Wörter, aber auch gestemmte Nominalchunks. Die Reihung der ähnlichen Dokumente erfolgt nach Auswertung der Vektoren, die meistens mittels des Kosinus Ähnlichkeitsmaßes auf deren Distanz geprüft werden, wodurch auch Teile von Nominalchunks berücksichtigt werden, in absteigender Reihenfolge um ein optimales Ergebnis zu liefern (vgl. Baeza-Yates und Ribeiro-Neto, 1999).

Die Autoren Li, Wong, Yuan, Li und Xia (2005) haben in Bezug auf Sätze anstatt von Dokumenten einen verbesserten ähnlichen Ansatz, das Critical Sentence Vector Model, vorgestellt, bei welchem zur Gewichtsberechnung der Sätze neben den normalen Wörtern auch Titelwörter, häufig vorkommende Wörter und Eigennamen berücksichtigt werden. Die Berechnung der Satzähnlichkeit basiert außerdem auf einem Vergleich der Subjekt-, Verb- und Objektstruktur und dem Vergleich der

längsten gemeinsamen Wortkette, wobei die Reihenfolge der Wörter außer Acht gelassen werden kann.

In Bezug auf die automatische Fragengenerierung wäre dieses Konzept einerseits für die Ermittlung von relevanten Textpassagen nützlich, indem man zusätzlich mit Query Termen arbeitet, andererseits können die Erkenntnisse dieses Modells auch dazu genutzt werden, um gegebene Antworten mit Referenzantworten zu vergleichen und demnach eine automatische Evaluierung von Antworten auf offene Fragen zu implementieren.

2.1.4.4 Rocchio Algorithmus

Der Rocchio Algorithmus gilt als einer der einfachsten Klassifikationsalgorithmen und kommt laut den Autoren Vinot und Yvon (2003) vor allem bei einer großen Klassenanzahl, wenn nicht nur das beste sondern die n besten Ergebnisse genommen werden und die Klassen ein großes Rauschen aufweisen, zum Einsatz.

Beim Trainieren des Klassifikators werden dabei auf eine Query Dokumentensets jeweils als positiv oder negativ eingestuft und so genannte Zentroiden Vektoren für die jeweilige Klasse aufgestellt. Im Endeffekt berechnet der Rocchio Algorithmus den Abstand eines Dokumentes zu den Zentroiden, der sowohl negativ als auch positiv bewerteten Dokumenten der Trainingsmenge und die Klasse mit dem kleinsten Abstand (vgl. Brückner, 2001).

2.1.4.5 Naive Bayes Klassifikator

Der naive Bayes Klassifikator basiert auf bedingten Wahrscheinlichkeiten beziehungsweise der Bayes-Formel und versucht vorherzusagen, ob ein Objekt einer bestimmten Klasse angehört. Als Klasse versteht man wiederum jene Objekte, die gewisse identische Eigenschaften aufweisen. Dabei wird jedem Objekt ein Merkmalsvektor zugeordnet und vorausgesetzt, dass die Objekte voneinander statistisch unabhängig sind (vgl. Hastie, Tibshirani und Friedman, 2009). Laut Brückner (2001) sind die empirischen Ergebnisse dieses Klassifikationsverfahren im Vergleich zu den anderen hier vorgestellten Klassifikationsalgorithmen eher schlecht.

2.1.4.6 k-Nearest-Neighbor

Beim k-Nearest Neighbor Algorithmus werden zu einem Objekt die k nächsten Nachbarn gesucht, wobei beim Trainieren die Merkmale der einzelnen Objekte in Vektoren und die jeweils zugehörigen Klassen gespeichert werden. Bei der Klassifikation werden die neuen Vektoren der Objekte mit den Trainingsvektoren verglichen und anhand dieses Vergleiches die k nächsten Objekte bestimmt und in die gleiche

Klasse wie jene eingeordnet (vgl. Brückner, 2001). Im Grunde genommen wird demnach die euklidische Distanz zwischen den Objekten ermittelt.

2.1.5 Bewertungskriterien

In diesem Subkapitel werden einige im Information Retrieval gängige, im Laufe dieser Arbeit noch öfters auftretende, Kriterien vorgestellt, die zur Bewertung von Termen, Dokumenten und Ähnlichem, im Vergleich zu der vollständig verfügbaren Menge dieser, verwendet werden.

2.1.5.1 Precision

Unter Precision versteht man laut den Autoren Baeza-Yates und Ribeiro-Neto (1999) einen Wert zwischen null und eins, der das Verhältnis von der brauchbaren zurück gelieferten zur gesamten zurück gelieferten Information angibt. Die Precision gibt also den Prozentsatz der relevanten Suchergebnisse, extrahierten Terme etc. an.

2.1.5.2 Recall

Der Recall ist ebenso eine Zahl zwischen null und eins, die das Verhältnis der relevanten zurück gelieferten Information zur gesamt verfügbaren relevanten Information angibt (vgl. Baeza-Yates und Ribeiro-Neto, 1999).

2.1.5.3 F-Measure

Die F-Measure ist das harmonische Mittel von der Precision und dem Recall und gibt gewissermaßen den besten Kompromiss zwischen diesen an. Um ein hohes harmonisches Mittel zu erreichen, müssen beide Werte sehr hoch sein (vgl. Baeza-Yates und Ribeiro-Neto, 1999).

2.1.6 NLP Anwendungsgebiete

Nachfolgend werden die wichtigsten Anwendungsgebiete im Bereich des Natural Language Processing vorgestellt, die einen hohen Bezug zu der Aufgabenstellung aufweisen und dahingehend Erkenntnisse über die prinzipielle Vorgehensweise liefern können.

2.1.6.1 Informationsextraktion

Unter Informationsextraktion versteht man, in Abhängigkeit der Kategorie, die Extraktion von relevanten Informationen, wobei hierbei verschiedene Schritte zu unterscheiden sind. Zuerst wird das Ziel verfolgt, aus unstrukturierten Daten, im vorliegendem konkreten Fall also Texten, die optimalerweise wohlgeformt sind, Informationen zu gewinnen. Die unstrukturierten Daten sollen eine Bedeutung erhalten und strukturiert werden, damit diese maschinell weiterverarbeitbar sind. Aus den strukturierten Daten werden anhand deren linguistischen Aufbaus semantische Informationen gesucht, jene als Grundlage für spezifische Informationsextraktion genutzt und die Ergebnisse daraus gewonnen werden erneut strukturiert und klassifiziert (vgl. De Busser, 2006).

Die Informationsextraktion wird demnach im Kontext der Aufgabenstellung dazu genutzt, Informationen aus dem unstrukturierten Text zu erfassen und in Abhängigkeit der Kategorie zu klassifizieren. Die als relevante eingestuften Wörter und Textpassagen dienen dann als Ausgangspunkt für die automatische Aufgabenerstellung.

2.1.6.2 Automatische Textzusammenfassung

Bei der automatischen Textzusammenfassung wird im Grunde genommen das gleiche Ziel wie bei der Aufgabenstellung dieser Arbeit, der automatischen Fragengenerierung, nämlich das Auffinden der aussagekräftigsten Textpassagen und Wörter im Text, verfolgt. Die Autorin Endres-Niggemeyer (2001) nennt als analoge Aufgabestellung das Abstracting, bei dem ebenso die wichtigsten Kernaussagen und deren Zusammenhänge im ganzen Text hervorgebracht werden sollen. Dabei prägen vor allem der pragmatische Kontext und der Zweck der Zusammenfassung das Ergebnis, also indirekt auch die Gewichtung der einzelnen verwendeten Methoden (vgl. Endres-Niggemeyer, 2001).

Die wesentlichsten Merkmale, die als Grundlage für die Analyse und die Satzauswahl genutzt werden, sind dabei in einem Textzusammenfassungssystem von Kupiec, Petersen und Chen (1999) die Satzlänge, Indikatorphrasen, die Absatzstruktur, Schlüsselwörter und Akronyme. Es werden demnach also jene Sätze in die engere Auswahl genommen, die eine angemessene Länge, wichtige Phrasen, häufige Inhaltswörter und Akronyme aufweisen beziehungsweise beinhalten und bevorzugt in den ersten oder letzten Absätzen vorkommen. Die letztendliche Übereinstimmungsrate dieses maschinellen Ansatzes mit den Ergebnissen von Menschen beträgt in etwa 35 Prozent (vgl. Endres-Niggemeyer, 2001).

Die Autoren von Ledeneva, Gelbukh und Garciá-Hernández (2008) unterscheiden zudem die *abstractive* und die *extractive* Textzusammenfassung. Erstere hat die Aufgabe den Kontext des Textes zu beschreiben, das Textverständnis zum

Ausdruck zu bringen und diesen in wenigen Worten zusammenfassen. Zweitere ist eine Zusammenfassung, die lediglich die wichtigsten Sätze ermitteln soll und diese eins zu eins übernimmt.

2.1.6.3 Textklassifikation

Der Autor Brückner (2001) schreibt über die wesentlichen Ziele der Textklassifikation, dass dabei vor allem Inhalte von größeren Textmengen automatisiert erfasst und kategorisiert werden sollen. Das Problem hierbei ist die Struktur und die Definition der Klassifikationsklassen, -typen und -ebenen, die von den Autoren Gansel und Jürgens (2007) näher beschrieben wird, für die vorübergehende Betrachtung jedoch wenig relevant ist.

Das prinzipielle Vorgehen bei der Textklassifikation beschreibt Brückner (2001) dabei folgendermaßen: Zuerst müssen den einzelnen Wörtern Merkmale verliehen werden, was zumeist durch Lemmatisierung und POS-Tagging realisiert wird. Zusätzlich wird dabei auch eine Wortbedeutungsanalyse durchgeführt, um mehrdeutigen Wörtern, je nach Kontext, einen eindeutigen Ausdruck verleihen zu können. Anschließend werden aus allen Merkmalen, die auf Wörtern oder N-Grammen basieren, die wichtigsten ausgewählt und daraus Modelle gebildet. Anhand von vorab definierten Klassifikationsverfahren, welche regelbasiert oder statistisch arbeiten, wird der Text letztendlich einer Kategorie zugewiesen (vgl. Brückner, 2001).

Auf den ersten Blick mag diese NLP Anwendung im Sinne der Aufgabenstellung dieser Arbeit als unpassend erscheinen, das Problem der Textklassifikation kann dabei aber einen wesentlichen Erfolgsfaktor darstellen, da die Unterscheidung von wichtigen und unwichtigen Merkmalen eines Wortes eine große Rolle spielt. Demnach sind diverse Gewichtungen sehr stark von Kategorien abhängig und beeinflussen die Auswahl von relevanten Informationen in hohem Maße.

2.1.6.4 Informationsbeschaffung

Unter Informationsbeschaffung (Information Retrieval) versteht man das inhaltsorientierte Suchen von Informationen anhand von Suchanfragen. Meistens ist damit das Suchen von Dokumenten in einer großen Dokumentenmenge, wie dies etwa beim WWW und Suchmaschinen der Fall ist, gemeint. Die Suchanfrage (query) des Nutzers ist oftmals unpräzise, woraus sich ein zusätzliches Problem ergibt, da jene erheblichen Einfluss auf die Qualität der ermittelten relevanten Information hat. Aus diesem Grund werden die Suchterme oftmals automatisch ausgebessert und wenn nötig auch erweitert (vgl. Baeza-Yates und Ribeiro-Neto, 1999).

Für die Suche nach Informationen beziehungsweise Dokumenten im gesamten Datenbestand werden vor allem vektorbasierte Modelle, boolsche Modelle und probabilistische Modelle eingesetzt. Die ermittelten Informationen werden anschließend gerankt, was hierbei ebenso ein essentieller Faktor, wie die query Formulierung ist, und dargestellt. Der Prozess des Suchens ist als iterativer Prozess anzusehen, da der Nutzer je nach Antwort des Systems die Suchparameter effizienter gestalten kann, sofern die gelieferten Informationen eine zu geringe Relevanz aufweisen (vgl. Baeza-Yates und Ribeiro-Neto, 1999).

2.1.6.5 Topic Segmentation

Unter Topic Segmentation versteht man laut den Autoren Moens, Angheluta und De Busser (2003) die Identifikation von Themen und Subthemen in einem Text. Einer der besten Ansätze geht dabei auf Hahn (1990), der drei verschiedene Muster eingeführt hat, zurück. Einerseits kann für jeden Absatz ein spezifisches Thema bestimmt werden, anderseits können aber auch Grenzen innerhalb von Sätzen, die einen Themenwechsel konstatieren, berechnet werden. Die dritte Möglichkeit ist die Ermittlung eines Themas durch die Subthemen die rundherum festgestellt wurden.

Der am meisten verbreitete Algorithmus zur Topic Segmentation ist der *TextTiling* Algorithmus von Hearst (siehe auch Kapitel 5.4), welcher wie nachfolgend funktioniert. Zu Beginn wird der Text tokenisiert, Stoppwörter entfernt, Verben und Nomen lemmatisiert und der Text in so genannte Pseudosätze fester, gleicher Länge unterteilt. Die einzelnen, sich überlappenden Blöcke werden danach in der lexikalischen Analyse auf deren Ähnlichkeit geprüft, indem das Skalarprodukt zweier Vektoren mit den beinhalteten Wörtern berechnet wird. Je größer dieser Wert, desto größer ist der Zusammenhang zweier Blöcke.

Zusätzlich wird jeder Block mit dem vorherigen Block verglichen und die Anzahl der normierten neu auftretenden Wörter als Maß für einen Abschnittswechsel herangezogen. Als Grenzen werden letztendlich genau jene Blöcke herangezogen, die einen sehr niedrigen Score erhalten haben und von zwei hochrangigen Blöcken umgeben sind (vgl. Hearst, 1997). Nach Detektion der Grenzen, die einen Themenwechsel anzeigen, werden die Themen durch einen Satz oder durch Schlüsselwörter in diesem Segment ausgedrückt (vgl. Moens, Angheluta und De Busser, 2003).

Weitere Algorithmen sind die von Labadié und Prince (2008) genannten C99 und Transeg Algorithmen, wobei ersterer auf dem Prinzip der lexikalen Kohäsion beruht und zweiterer distanzbasiert arbeitet. Die Autoren Blei und Moreno (2001) stellen einen Ansatz vor, der, unter Zuhilfenahme des Viterbi Algorithmus, ein Aspect Hidden Markov Model implementiert und die thematischen Zusammenhänge der einzelnen Wörter darstellt.

Eines der wesentlichen Einsatzgebiete der Topic Segmentation ist grundsätzlich die Textzusammenfassung, indem nach Detektion der Grenzen, die einen Themenwechsel anzeigen, die Themen durch einen Satz oder mehrere Schlüsselwörter in diesem Segment ausgedrückt werden (vgl. Moens, Angheluta und De Busser, 2003). Die gewonnen Erkenntnisse über die Satzgrenzen könnten jedoch dazu genutzt werden, Referenzantworten auf offene Fragen automatisch zu bestimmen. Es müsste nur jener Abschnitt gesucht werden, der den erfragten Term beinhaltet und die beste Bewertung erhalten hat. Somit würde die Referenzantwort, sofern der Algorithmus optimal funktioniert, nur aus tatsächlich relevanten Informationen bestehen.

2.2 Assessment

In diesem Abschnitt der vorliegenden Arbeit wird auf Assessment, welches, vor allem aufgrund der stetig wachsenden E-Learning Inhalte und dem daraus resultierenden Bedürfnis nach Selbstüberprüfung, sowie den mannigfaltigen Möglichkeiten eines Einsatzes in jeglicher Form von Lehre, an Bedeutung gewinnt, eingegangen. Die wesentlichen Merkmale von Assessment, im Fokus der Aufgabenstellung, sind unteren anderem die Planung, die Diskussion, die Erstellung von Übereinstimmungen und Maßsystemen, sowie die Reflexion und Analysen der Lernziele (vgl. Gütl und AL-Smadi, 2008).

Die Aufgabenstellung der Entwicklung von Konzepten zur automatischen Fragengenerierung ergibt sich implizit aus den Anforderungen von Assessment beziehungsweise dem Einsatz von Assessment-Systemen, da jene einen enormen Zeitaufwand in Bezug auf die Erstellung von wissensüberprüfenden Aufgaben und Tests verursachen (vgl. Mitkov und Ha, 2003). Von daher wäre es in weiterer Folge wünschenswert computerunterstützt, also soweit als möglich vollautomatisch, die wesentlichen semantischen Inhalte eines Textes zu erfassen und dahingehend zu extrahieren, dass eine automatische Erstellung von Aufgaben möglich ist.

Die Autoren Rus, Cai und Graesser (2007) merken darüber hinaus an, dass in mehreren psycholinguistischen Studien bewiesen wurde, dass das Stellen von Fragen, wobei tiefgründige Fragen zu bevorzugen sind, den Lernfortschritt signifikant steigen lässt. Es spielt dabei eine geringe Rolle, ob die Fragen vom Lernenden oder Lehrenden kommen, denn dieser Effekt ist immer beobachtbar. Diese Tatsache verdeutlicht zusätzlich den Nutzen und die Sinnhaftigkeit von Assessment.

Nachfolgend wird kurz auf Assessment im Bereich E-Learning eingegangen, wobei einige richtungweisende Ansätze über die Entwicklung von Assessment Systemen dargelegt werden.

2.2.1 Assessment im Bereich E-Learning

Roberts (2006) unterteilt Assessment im Bereich vom E-Learning, das eine der größten Motivationsquellen für diese Arbeit darstellt, in Self-Assessment, Peer-Assessment und Group-Assessment und verdeutlicht die unterschiedlichen Zielsetzungen und Bedürfnisse jener. Self-Assessment wird durch aktives Lernen, kritische Reflexionen, Protokollieren der Fortschritte und durch Vorschläge zur eigenen Notengebung charakterisiert. Dies verdeutlicht das Bedürfnis nach Assessment Systemen, die, wie sich Laufe dieser Arbeit noch zeigen wird, Möglichkeiten für all jene Eigenschaften bieten. Auch das Konzept von Peer-Assessment, also die gegenseitige Beurteilung, kann dahingehend Vorteile bringen, dass Feedback von anderen Lernenden, die ähnliche Sichtweisen vertreten, eventuell besser aufgenommen werden kann, als jenes von Lehrenden. Group-Assessment, also die Beurteilung von Gruppen von Lernenden spielt in Bezug auf die Aufgabenstellung eine sehr untergeordnete Rolle, weshalb dieses Konzept für die weiterführenden Überlegungen verworfen wird (vgl. Roberts, 2006).

Die Autoren Gütl und AL-Smadi (2008) unterscheiden darüber hinaus bei E-Assessment die zwei Arten des formativen und des summativen Assessments. Formatives Assessment ist demnach ein Teil des Lernprozesses, der sowohl dem Lernenden als auch dem Lehrenden Feedback geben soll, das sofort interaktiv in den Lernprozess miteinbezogen werden kann und somit den Lernprozess optimiert. Summatives Assessment ist die Überprüfung und die Evaluierung des Wissens am Ende der Lernperiode.

2.2.2 Fragetypen

In diesem Subkapitel werden die wichtigsten Fragetypen und deren Aufbau, sowie deren Sinnhaftigkeit und Zweck erläutert. Im Wesentlichen wird zwischen offenen und geschlossenen Fragen differenziert. Unter offenen Fragen versteht man Fragen, auf die eine beliebige natürlich sprachliche Antwort gegeben werden kann. Im Gegensatz dazu sind bei geschlossenen Fragen die Antwortmöglichkeiten vorkonstruiert und vorgegeben.

Die wichtigsten Kriterien bei der Auswahl eines Fragetypes sind laut den Autoren McKenna und Dougherty Stahl (2009) unter anderem der Hintergrund der Frage, da die verschiedenen Typen andere, jedoch nicht empirisch belegte Sichtweisen hervorrufen, die Bewertungskriterien welche zum Einsatz kommen sollen und wie groß das erfragte Textkorpus ist. Wiggins und McTighe (2005) sehen dies ähnlich und meinen, dass die einzusetzenden Assessment Methoden vom Wissensstand und den Zielen des Lehrplans abhängen. Dabei zielen offene Fragen eher darauf ab zu einem Thema die Kernkonzepte und grundlegenden Ideen dahinter zu erfahren,

wohingegen Fragen mit vorgegebenen, konstruierten Antworten auch Bereiche abdecken können, mit denen die Lernenden weniger vertraut sind.

Bei derartigen Multiple Choice Fragen gilt es geeignete Distraktoren zu ermitteln, die einerseits ein ähnliches Konzept, wie das erfragte, darstellen, aber andererseits, ohne genügend Hintergrundwissen, keinen Aufschluss darüber geben, ob jene richtig oder falsch sind.

Single Choice Fragen können generiert werden, indem die Satzstruktur des zu erfragenden Satzes verändert oder eine Phrase gegen eine andere ausgetauscht wird. Ersteres kann durch die vom Autor Chen (2003) in dem Buch *English Inversion* beschrieben Problematik der Inversion von Sätzen erreicht werden. Es wird dabei aufgezeigt, dass es keine einfachen Regeln für eine Inversion gibt, sondern dass es, vor allem bei komplexen Sätzen, mehrdeutige Möglichkeiten der Umsetzung möglich sind. Die Vorgehensweise zum Transformieren eines Aussagesatzes in einen Fragesatz ist das Vertauschen der Subjekt-Verb-Struktur.

Den Autoren Wiggins und McTighe (2005) nach stehen beim Prozess des Aufgabenerstellens im ersten Schritt aus der Sicht der Lehrenden immer die Merkmale, die es zu erfragen gilt und welche im nächsten Schritt auch nach gewissen Charakteristiken zu beurteilen sind, im Vordergrund[2]. Darüber hinaus ist zu berücksichtigen, dass jene Charakteristiken Aufschluss über die Zuverlässigkeit und Validität der Aufgaben geben.

2.2.3 E-Assessment Systeme

In diesem Unterkapitel geht es weniger darum, die vollständige Funktionsweise von Assessment Systemen vorzustellen, sondern es sollen anhand einiger kleiner Beispiele die Anforderungen, der Stellenwert und der Nutzen jener dargelegt werden. Zur Veranschaulichung der detaillierten Vorgehensweise werden im Kapitel 3.3 einige Beispiele angeführt und näher beschrieben.

Es kann ohne Einbeziehung der Lernzielvorgabe keine generelle Aussage getroffen werden welche Methoden ein E-Assessment System bieten sollen, da jene auf unterschiedliche Bereiche abzielen. Darüber hinaus unterscheidet man die so genannten *fixed response* und die *free response* Systeme, wobei erstere eher auf reines Wissen an sich und zweitere auf Meta Skills und auch die Fähigkeit von Ausdrucksweisen abzielen (vgl. Gütl und AL-Smadi, 2008).

[2] *"What kinds of evidence do we need to find hallmarks of our goals, including that of understanding?" (*Wiggins und McTighe, 2005, *S. 150)*

Einige allgemein gültige Regeln die es beim Entwurf eines E-Assessment Systems immer zu beachten gilt beschreiben Gütl und AL-Smadi (2008): Ein derartiges System soll dem Stand der Technik und den sich daraus ergebenden Möglichkeiten entsprechen, Sicherheit und Privatsphäre bieten, Feedback geben, da jenes essentiell für den Lernfortschritt ist, den gängigen Standards folgen und Austauschbar sein, ein verständliches Benutzerinterface haben, flexibel sein sowie automatisierte und einstellbare Abläufe implementieren. Weiters ist es natürlich wünschenswert, dass Assessment Systeme der Vollständigkeit halber auch Analysen in Bezug auf gegebene Antworten durchführen, um die Fehler und Schwächen der Nutzer effizient aufzuzeigen und dahingehend ein optimales Feedback generieren. Als Standard für die Aufgaben bei E-Assessment Systemen sei stellvertretend der IMS QTI Standard (vgl. QTI, 2009, siehe auch Kapitel 5.3) erwähnt, der ein XML Schema mit Tags für alle relevanten Informationen zu den Fragen, den Antworten, der Bewertung etc. implementiert.

Einen interessanten Ansatz zur Thematik des Peer Assessments liefern die Autoren Gütl, AL-Smadi und Kappe (2009), die ein System entwickelt und im Rahmen einer Lehrveranstaltung, an der auch der Autor der vorliegenden Arbeit teilnehmen hat dürfen, getestet haben. Dabei hatten Studenten zu Beginn Fragen zu einem vorab gelesenen Text zum Thema Dokumentklassifikation zu beantworten und in weiterer Folge die Antworten von anderen Studenten zu verifizieren und die Unklarheiten herauszuarbeiten, indem die Antworten durch verschiedene Tags zu kennzeichnen waren. Die Ergebnisse dieser Studie besagen, dass man Wissen neben dem eigentlichen Lernen und dem Lernen mit Hilfe von Referenzantworten durchaus in ähnlichem Maße durch Peer Assessment erlangen kann und jene Ansätze bei zukünftigen Systemen in der Lehre sicher ihren berechtigten Platz einnehmen werden.

2.3 Zusammenfassung

In diesem Kapitel wurden zu Beginn einige grundlegende und allgemeine Begriffe zum Thema Natural Language Processing dargelegt und es wurde aufgezeigt, welche breit gefächerten, komplexen Problemstellungen sich in diesem Gebiet ergeben. Danach wurde im Fokus der Aufgabenstellung dargelegt, welche Schritte durchzuführen sind, um beliebige Texte einerseits syntaktisch, pragmatisch und semantisch, andererseits aber auch statistisch zu analysieren und weiterzuverarbeiten, um eine Basis dafür zu schaffen, wichtige Inhalte zu erkennen und zu extrahieren.

Im zweiten Abschnitt dieses Kapitels wurde der Begriff Assessment im Kontext von E-Learning, einer wesentlichen Motivation dieser Arbeit, herausgearbeitet und

aufgezeigt, dass zwischen Self-, Peer- und Group-Assessment zu unterscheiden ist. Darüber hinaus wurden kurz die einzelnen Fragetypen abgehandelt und die Anforderungen als auch der Nutzen von E-Assessment Systemen aufgezeigt, da es ja mitunter ein Ziel dieser Arbeit ist, einen Teil eines derartigen Systems zu entwickeln. Es hat sich gezeigt, dass derartige Systeme im besten Fall dem Stand der Technik entsprechen, Sicherheit bieten, die Privatsphäre wahren, flexible und austauschbare Module beinhalten, eine verständliche Benutzeroberfläche haben, automatisiert Feedback geben und allen Standards entsprechen.

Im nachfolgenden Kapitel werden der Forschungsstand und bewährte als auch innovative Umsetzungen der dargelegten Methoden und Konzepte vorgestellt, um in weiterer Folge daraus Schlüsse für die Aufgabenstellung ziehen zu können.

3 *Aktuelle Anwendungen und Forschungsstand*

In diesem Kapitel soll aufgezeigt werden, wie die grundlegenden Konzepte, welche in Kapitel 2 erarbeitet wurden, in der Praxis eingesetzt werden und welche Forschungsansätze dahingehend verfolgt werden. Natürlich kann aufgrund der vielfältigen Ansätze, die in diesem Gebiet aufgrund der steigenden Möglichkeiten in Bezug auf die Rechenleistung und anderen technisch bedingten Möglichkeiten existieren, nur ein Auszug derer angegeben werden. Der Fokus liegt dabei auf der Extraktion von Termen und Konzepten anhand verschiedener Ansätze, der automatischen Generierung von Fragen und Aufgaben sowie automatischem Assessment.

3.1 Automatische Term- und Konzeptextraktion

Die automatische Term und Konzeptextraktion ist für viele natürlichsprachliche Anwendungen ein Schlüsselkonzept und die Grundlage für die Item Auswahl der automatischen Aufgabenerstellung. Nachfolgend werden deshalb Methoden beschrieben die, im Kontext anderer NLP Anwendungen, vor allem dem Information Retrieval, der Themenklassifikation und der Textzusammenfassung, verwendet werden, für die Entwicklung von Konzepten für die Wort und Termauswahl bei der Fragengenerierung jedoch eine große Rolle spielen.

Der klassische Ansatz zur automatischen Textzusammenfassung, die eine Termextraktion zu Grunde hat, basiert auf den vom Autor Stock (2007) genannten Ansätzen von Luhn aus dem Jahre 1958. Dabei werden Wortgewichtungen eingeführt, da laut Luhn jene Häufigkeit ein Indiz für die Signifikanz eines Wortes ist. Dabei werden jedoch nicht die häufigsten Wörter, sondern jenen die eine mittlere Häufigkeit aufweisen als signifikant eingestuft, die Signifikanzverteilung folgt demnach einer Gaußglockenverteilung.

3.1.1 Domain-specific Keyphrase Extraction

Die Autoren Frank, Paynter, Witten, Gutwin und Nevill-Manning (1999) stellen in ihrer Arbeit eine Möglichkeit vor, wichtige Schlagwörter und Phrasen in einem Dokument anhand eines Naiven Bayes Klassifikators zu bestimmen. Darüber hinaus haben sie aufgezeigt, dass die Performance der Klassifikation wesentlich von der Art der Trainingsdokumente abhängt, worauf nachfolgend noch näher eingegangen wird.

Zu Beginn des, zu diesem Ansatz begleitend implementierten, KEA (Keyphrase Extraction Algortihm) Algorithmus werden im Dokument alle Zahlen ausgeklammert

und in Abhängigkeit der Satzzeichen alle möglichen Phrasen der maximalen Länge von drei gebildet. Jene werden vorerst als Kandidaten eingestuft. Im nächsten Schritt werden jene Phrasen, die mit einem Stoppwort beginnen oder enden und welche vorwiegend aus Eigennamen bestehen, eliminiert. Alle übrigen Kandidaten werden auf deren Grundform reduziert und jene, die nur einmal im Text auftreten, von der Liste gestrichen (vgl. Frank, Paynter, Witten, Gutwin und Nevill-Manning, 1999).

Die tatsächliche Entscheidung ob eine Phrase eine relevante Schlüsselphrase ist hängt im Wesentlichen von zwei Faktoren ab. Einerseits bestimmen die Termhäufigkeit und die invertierte Dokumentenfrequenz (tf-idf) die Klassifikation, andererseits wird die Distanz, welche das erste Auftreten der Phrase geteilt durch alle Wörter des Dokumentes verkörpert, berücksichtigt. Die Bayes Klassifikation arbeitet anhand der Formel in Abbildung 2, wobei ersichtlich ist, dass die Wahrscheinlichkeit für die Einordnung als relevante Schlüsselphrase von den unabhängigen, normierten Wahrscheinlichkeiten für die tf-idf, die Distanz und die Einstufung als relevant im ersten Schritt des Algorithmus abhängt. Bei gleich gut bewerteten Phrasen wovon aufgrund der Vorgabe zu viele klassifiziert sind, wird die tf-idf Bewertung bevorzugt, wenn eine Phrase jedoch eine Subphrase einer anderen ist, so wird die höhere Gesamtbewertung herangezogen (vgl. Frank, Paynter, Witten, Gutwin und Nevill-Manning, 1999).

$$\Pr[key|T,D] = \frac{\Pr[T|key] \times \Pr[D|key] \times \Pr[key]}{\Pr[T,D]}$$

Abbildung 2: KEA Klassifikation (vgl. Frank, Paynter, Witten, Gutwin und Nevill-Manning, 1999)

Die Evaluierung der klassifizierten Phrasen hat laut den Autoren Frank, Paynter, Witten, Gutwin und Nevill-Manning (1999) verdeutlicht, dass die automatisch bestimmten Phrasen großteils mit jenen von den Autoren gelieferten übereinstimmen und dass die Performance des Algorithmus mit der Größe der Trainingsdaten skaliert. Am besten waren die Resultate bei etwa 50 Trainingsdokumenten, wobei weitere Tests mit verschiedenen Settings zeigten, dass, sofern die Anzahl der verwendeten Schlüsselphrasen beim Trainieren in die obig genannten Kalkulationen mit einbezogen wird, bei Trainings- und Klassifikationsdokumenten aus der gleichen Kategorie die Ergebnisse wesentlich verbessert werden konnten.

3.1.2 Pattern Extraction

Einen Ansatz zur Unterscheidung von so genannten *subjective sentences* und *objective sentences* beziehungsweise der Bestimmung von *subjective nouns* haben Riloff, Wiebe und Wilson (2003) in deren Arbeit *Learning Subjective Nouns using Extraction Pattern Bootstrapping* vorgestellt. Dabei werden anfangs mittels zweier

Bootstrapping Algorithmen *subjective nouns*, also subjektive Informationen, bestimmt, indem Extraktionsmuster benutzt werden und anschließend ein naiver Bayes Klassifikator mit diesen trainiert. Als Bootstrapping Algorithmen wird Meta-Bootstrapping und die Basilisk Variante eingesetzt.

Als Annotationsschema kommt ein für derartige Zwecke speziell entwickeltes zum Einsatz, das wesentlich umfangreicher ist als herkömmliche Schemata (siehe auch Kapitel 5.1.1). Dabei sind unter anderem Tags für *private states*, die als niedrig bis extrem hoch eingestuft werden können, enthalten. Die Klassifikationsdaten bestanden aus 109 Nachrichtendokumenten mit 2197 Sätzen aus dem U.S. Foreign Broadcast Information Service, die Trainingsdaten waren ebenfalls ein Teil dieser. Die Annotation von den 12 Trainingsdokumenten mit besagtem Schema erfolgte durch zwei Personen auf händische Art, wobei die Annotationen anschließend abgeglichen wurden. Die Sätze wurden im nächsten Schritt in drei Kategorien, die *subjective, objective* und *borderline sentences* unterteilt, wobei ein subjektiver Satz mindestens ein *private state* mit medium, ein borderline Satz nur *private states* mit low beinhalten darf (vgl. Riloff, Wiebe und Wilson, 2003).

Die beiden Bootstrapping Algorithmen arbeiten grundsätzlich nach Mustern wie „<subject> was hired“, woraus sich ergibt, dass Wörter die in das subject Schema passen einer gewissen semantischen Kategorie angehören. Anhand einiger weniger syntaktischen Muster und den zugehörigen *seed* Wörtern (händisch, semantisch kategorisierte Wörter) extrahieren der Meta-Bootstrapping Algorithmus alle möglichen Muster im Text und bewertet die einzelnen Muster anhand der Anzahl der darin enthaltenen seed Wörter. Danach werden alle Hauptwörter die vom besten Muster beinhaltet sind kategorisiert und in ein semantisches Wörterbuch eingetragen. Dieses Vorgehen wird iterativ wiederholt und jeweils das aktuelle Wörterbuch herangezogen. Am Ende werden nur die besten fünf Wörter im Wörterbuch beibehalten und der Vorgang erneut iterativ wiederholt (vgl. Riloff, Wiebe und Wilson, 2003).

Basilisk arbeitet im ersten Schritt identisch wie Meta-Bootstrapping, im zweiten Schritt jedoch werden alle Wörter die einem beliebigen Muster angehören bewertet. Dazu wird nicht nur die Anzahl der Beinhaltungen, sondern auch die kollektive Verwendung in den Mustern miteinbezogen. Die besten zehn Wörter werden semantisch eingeordnet, in das Wörterbuch geschrieben und der Vorgang erneut iterativ mit den seed Wörtern und den Wörtern im Wörterbuch als Ausgangspunkt gestartet. Im Gegensatz zu Meta-Bootstrapping berücksichtigt Basilisk anstatt des besten Musters die kollektive Information von allen extrahierten Mustern (vgl. Riloff, Wiebe und Wilson, 2003).

Als Seed Wörter dienten die 20 besten von 850, in einem anderen Test als subjektiv eingestufte Wörter aus 950 alternativen Texten der FBIS Sammlung. In 400 Iterationen wurden mit beiden Bootstrapping Algorithmen 2000 subjektive Wörter

extrahiert und händisch als *weak* oder *strong subjective* eingestuft. Das endgültige Wörterbuch umfasst 825 Wörtern beim Basilisk und 522 Wörter beim Meta-Bootstrapping, wobei die Überlappung der gleich eingestuften Wörter mittelmäßig ausfiel. Der naive Bayes Klassifikator arbeitete mit Subjektivitätsfeatures, also einer Einstufung des Subjektivitätsmaßes von Hauptwörtern, WBO Features, also Listen von sowohl positiv als auch negativ mit Subjektivität korrelierenden Wörtern, einigen manuell hinzugefügten Features aus anderen Testreihen und der Satzstruktur, wobei die einzelnen Sätze nach der Dichte von Subjektivitäts- und Objektivitätsmerkmalen und deren Länge bewertet wurden. Die Auswertungen der Ergebnisse finden sich in Tabelle 1 und zeigen, dass die Kombination obig genannter Kriterien die besten Resultate mit einer Genauigkeit von 76.1 Prozent lieferte (vgl. Riloff, Wiebe und Wilson, 2003).

	Acc	Prec	Rec	
(1)	76.1	81.3	77.4	WBO+SubjNoun+ manual+discourse
(2)	74.3	78.6	77.8	WBO+SubjNoun
(3)	72.1	76.0	77.4	WBO

Tabelle 1: Evaluierung der Bootstrapping Methode (vgl. Riloff, Wiebe und Wilson, 2003)

3.1.3 Lexikalische Ketten

Die Autoren Song, Han und Rim (2004) stellen in ihrer Arbeit *A Term Weighting Method Based on Lexical Chains* eine Methode vor, um Terme nach deren lexikalischer Kohäsion zu gewichten und die Erkenntnisse daraus für automatische Zusammenfassungen nutzen zu können. Es werden dabei sowohl Assoziationen zwischen den einzelnen Wörtern als auch Charakteristiken von WordNet ausgewertet. Aus den Wörtern, die zueinander einen semantischen Zusammenhang besitzen, werden Gruppen und so genannte lexikalische Ketten gebildet. Eine Grundannahme für dieses Vorgehen ist, dass jene Wörter, die viele semantische Abhängigkeiten im Text hervorrufen, für den Inhalt sehr relevant sind.

Bei diesem Ansatz sind zwei wesentliche Problemstellungen zu berücksichtigen. Einerseits sind jene lexikalischen Ketten oftmals mehrdeutig und hängen von den Bedeutungen der einzelnen Wörter ab. Haben jene also mehrere Bedeutungen, so kann auch die lexikalische Kette mehrere Bedeutungen besitzen. Andererseits muss eine Methode gefunden werden, um zu unterscheiden, welche der extrahierten lexikalischen Ketten in Bezug auf die Thematik relevant sind. Laut Song, Han und Rim (2004) werden diese zwei Problemstellungen meist mit heuristischen Ansätzen basierend auf der Anzahl und der Art von den Beziehungen der Ketten gelöst.

Die Bildung lexikalischer Ketten erfolgt analog dem Ansatz von Barzilay und Elhadad (1997), welche die stärksten Ketten und darauf aufbauend die besten Sätze

eines Textes extrahieren können. Wohingegen bei deren Berechnung keine Mehrdeutigkeiten ausgewertet werden, wird bei dem Ansatz von Song, Han und Rim (2004) zusätzlich eine Bewertungsfunktion eingeführt, die abschätzen soll, inwieweit die Beziehung von Wörtern richtig sein kann. Die Scoring Funktion besteht aus drei Teilen, dem *word association score*, der *depth in wordnet hierarchy* und dem *semantic relation weight.*

$$fs(w_1, w_2, r) = Assoc(w_1, w_2) \times DepthScore(w_1, w_2) \times \mathrm{Re}\, lationWeight(r)$$

Abbildung 3: Bewertung lexikalischer Ketten (vgl. Song, Han und Rim, 2004)

In Abbildung 3 ist jene Formel, anhand derer die einzelnen Wörter einer Kette bewertet werden dargestellt. Der *word association score* verdeutlicht den Zusammenhang zweier Wörter anhand der Anzahl des gemeinsamen Auftretens dieser beiden in WordNet. Die *depth in wordnet hierarchy* beschreibt das Produkt der Tiefe der beiden Wörter in WordNet, wobei die Bewertung höher wird, je tiefer sich die Synsets in der WordNet Hierarchie befinden. Es wird nämlich davon ausgegangen, dass jene tiefer gelegenen Wörter spezifischer sind und weniger Bedeutungen haben. Das *relation weight* hängt jeweils von der Art der Beziehung in WordNet ab. Wörter, die zumindest das zweite Mal auftreten erhalten das Gewicht von eins, Synonyme 0.2, Antonyme 0.3, Meronyme und Holonyme 0.4, Hyperonyme und Hyponyme 0.2 und Geschwister 0.05. Das endgültige Gewicht einer lexikalischen Kette wird nachfolgend über die Summe aller obig errechneten Wortpaare zusammengesetzt. Bei Mehrdeutigkeiten von Ketten wird immer jene mit dem höchsten Score ausgewählt (vgl. Song, Han und Rim, 2004).

Für die Termgewichtung, die letztendlich für die Auswahl der besten Sätze zeichnend ist, werden zu Beginn drei Annahmen getätigt. Zum einen müssen immer alle Teile einer Kette zu demselben Konzept gehören. Darüber hinaus wird angenommen, dass die Anzahl von Beziehungen von Wörtern in lexikalischen Ketten ein Maß für die Wichtigkeit dieser ist. Zusätzlich wird einem Wort innerhalb einer Kette mehr Wichtigkeit zugesprochen, wenn es mehrere Beziehungen zu anderen Wörtern innerhalb dieser Kette aufweist. Diese drei Faktoren werden in die so genannte *concept frequency* zusammengefasst und für die Berechnung der Satzrelevanz herangezogen, indem die Summe aller *concept frequencies* eines Satzes berechnet wird (vgl. Song, Han und Rim, 2004).

Die Evaluierung dieses Gesamtkonzept wurde anhand von zwanzig zufällig ausgewählten Dokumenten aus der DUC 2001 Sammlung durchgeführt. Als Vergleich wurden zwei Systeme herangezogen die im ersten Fall die einfache Termhäufigkeit beziehungsweise im zweiten Fall die Termhäufigkeit in Bezug auf die invertierte Dokumentenfrequenz für die Satzauswahl herangezogen haben. Dabei wurde aufgezeigt, dass beispielsweise die R-Precision um circa fünf Prozent

besser ist als bei den beiden alternativen Methoden. Fehleranfällig ist das vorliegende Konzept jedoch bei jeglichen Zeitangaben und Eigennamen, was aber auf die beschränkten Möglichkeiten diesbezüglich in WordNet zurückzuführen ist (vgl. Song, Han und Rim, 2004).

3.1.4 Topic Based Summarization

Der Autor Saggion (2005) stellt ein automatisches Zusammenfassungssystem vor, dass die Satzrelevanz, die Satzposition, die Ähnlichkeit des Satzinhaltes zum Titel, die Satzähnlichkeit zum ersten Paragraphen eines Dokumentes sowie die Termverteilung und Eigennamen berücksichtigt. Dazu wird vorerst ein Vector Space Model mit allen Wörtern und den zugehörigen Gewichten, die sich aus der Auftrittshäufigkeit und der invertierten Dokumenten Frequenz (idf) zusammensetzen, aufgestellt. Um die Ähnlichkeit zwischen einzelnen Textteilen zu berechnen wird das Kosinusähnlichkeitsmaß verwendet und zu jedem Cluster die Zentroidenrepräsentation ermittelt (vgl. Saggion, 2005).

Zusätzlich werden N-Gramme von den einzelnen Textfragmenten dazu genutzt, um die Ähnlichkeit zwischen diesen zu validieren und eine Redundanz zu gewährleisten. Außerdem werden, dem Ansatz von Marcu (1999) folgend, sukzessiv Sätze, die dem Abstract nicht ähneln ausgeklammert und nicht weiter in die Auswertung miteinbezogen. Nachfolgend werden für jedes Dokument die Themenbeschreibung im Vector Space Model implementiert und eine Dokumentenanalyse durchgeführt. Dabei werden die einzelnen Sätze mit allen wichtigen Informationen annotiert, und mittels der invertierten Dokumenten Frequenz in das Vector Space Model integriert. Danach werden jene Vektoren mit deren Kosinus Ähnlichkeitsmaß versehen und ein Schwellenwert für jeden Cluster berechnet. Jeder Satz mit einer überdurchschnittlichen Ähnlichkeit kommt letztendlich, sofern dieser sich von den anderen Sätzen in der Liste unterscheidet, auf die so genannte Kandidatenliste, bis sich die gewünschte Kompression erreicht ist (vgl. Saggion, 2005).

Die daraus entstehenden Zusammenfassungen wurden anhand von fünf Kriterien evaluiert. Die generierten Sätze dürfen keine internen Formatierungen, Fehler in der Groß- und Kleinschreibung sowie grammatikalische Fehler beinhalten. Darüber hinaus sollen Wiederholungen vermieden werden, referenzierende Phrasen zuordenbar sein, die ausgewählten Sätze den wichtigsten Inhalt widerspiegeln und richtig strukturiert sein. Bewertet wurden die erzeugten Zusammenfassungen mittels dem automatischen Recall-Oriented Understudy for Gisting Evaluation (ROUGE) System, das im wesentlichen den Recall der N-Gramme beurteilt, und der manuellen Pyramiden Methode, bei der anhand der Pyramidenformel ausgewertet wird, ob die so genannten *content units*, also die wichtigsten Informationsträger, in der Zusammenfassung vorkommen. Die Zusammenfassungen wurden mit einem ROUGE

Score von 24 aus 33 und einem Pyramiden Score von 13 aus 25 bewertet (vgl. Saggion, 2005).

3.1.5 CorePhrase

Ein Ansatz zur Extraktion von *key phrases* in einer Dokumentenmenge oder Clustern von Textfragmenten ohne jegliche Kenntnis über das Dokument beziehungsweise den Text wird von Hammouda, Matute und Kamel (2005) vorgestellt. Der CorePhrase genannte Algorithmus liefert eine Kandidatenliste indem die Dokumente anhand eines Indizierungsmodells in ein Graphenmodell, auch Document Index Graph (DIG) genannt, übergeführt werden. Jener Graph wird kumulativ aufgebaut, wobei jeder Subgraph eines Dokumentes in den bestehenden Graphen integriert wird. Es werden also jene Phrasen als Kandidaten ausgewählt, die im Prinzip Schnittpunkte zwischen den verschiedenen Dokumenten darstellen.

Im nächsten Schritt werden allen Wörtern auf der Kandidatenliste die Features *document frequency df*, die Anzahl der Dokumente die die Phrase beinhalten in Relation zu allen Dokumenten, *average weight w*, das Durchschnittsgewicht eines Wortes, wobei Titel unter Überschriften bevorzugt werden*, average phrase frequency pf*, dem Gewicht eines Wortes geteilt durch die Anzahl der Wörter, und die *average phrase depth d*, die den Ort des ersten Vorkommen innerhalb eines Dokumentes darstellt, zugewiesen. Die Bewertung der einzelnen Phrasen geschieht anhand der Formel in Abbildung 4, die eine Erweiterung eines ersten Prototyps darstellt, und durch Aufsummierung aller Wortgewichte (vgl. Hammouda, Matute und Kamel, 2005).

$$\mathrm{score}(p) = (\sqrt{w \cdot pf} \cdot d^2) \times -\log(1 - df)$$

Abbildung 4: Bewertung der Phrasen des CorePhrase Algorithmus, (vgl. Hammouda, Matute und Kamel, 2005)

Die Evaluierung des CorePhrase Algorithmus wurde mittels Dokumentenclustern die aus sechs Suchanfragen gebildet wurden und mittels einer Dokumentensammlung des Intranets der Universität von Waterloo durchgeführt. Als Grundlage für die Analysen und Vergleiche wurden von allen Dokumenten die zentroiden Vektoren aufgestellt und die besten Phrasen herausgesucht. Anschließend wurden als Vergleichskriterien der Overlap, ein Ähnlichkeitsmaß der extrahierten zu den vorab definierten *key phrases*, und die Precision, die in diesem Fall angibt, wie hoch das Ranking der besten Phrase je Themengebiet einzustufen ist. Zusammenfassend lässt sich sagen, dass die erzielten Ergebnisse laut Hammouda, Matute und Kamel (2005) besser sind als jene die durch zentroide Vektoren ermittelt wurden und der Einfluss der so genannten *phrase depth* relativ groß ist. Die Verwendung der einzel-

nen Wortgewichte verbessert die Ergebnisse, ohne Berücksichtigung der Wortgewichte wurden jedoch bessere deskriptive Phrasen berechnet.

3.1.6 Discourse Structure

Die Autoren Cristea, Postolache und Pistol (2006) stellen einen Ansatz vor, der Sätze in eine Baumstruktur transformiert und daraus aufgebaut, Referenzketten auszuwerten. Dabei wird der Text eingangs POS getaggt, ein syntaktischer Parser ausgeführt und Sätze in so genannte *elementary discourse units (edus)* segmentiert sowie jene Sätze in genannte *elementary discourse trees (edts)* übergeführt. Zusätzlich werden Noun Chunks und Anaphern detektiert und in den *edus* gespeichert. Danach wird ein iterativer Prozess ausgeführt, wobei für jeden Iterationsschritt alle zu diesem Zeitpunkt bestimmten *edts* gespeichert werden. Das heißt, dass jede Iteration eine andere Interpretation des Textes darstellt.

In einem Iterationsschritt werden alle edts, die auf den nächsten Satz verweisen, in allen möglichen Positionen eingesetzt. Eine besondere Rolle spielen cue Phrasen, auch *marker* genannt, die Verknüpfungen zwischen einzelnen edus darstellen, die also mindestens zwei Argumente miteinander in Beziehung bringen. Um diese Beziehungen auszuwerten wurden manuell Muster gebildet, wobei anzumerken ist, dass manche cue Phrasen nicht eindeutig sind. Diesen Phrasen, die ihrerseits eine edu in der edt Baumstruktur darstellen, werden nun die ausgewerteten Argumente zugewiesen und die Baumtiefe demnach erhöht (vgl. Cristea, Postolache und Pistol, 2006).

Zusätzlich werden *dummy marker* eingesetzt wenn keine cue phrase zur Verfügung steht, es müssen also in jedem Satz bei n edus n – 1 *marker* existieren. Wie die dadurch entstehenden Ketten genau ausgewertet werden wird nicht genauer Beschrieben, da dies den Rahmen dieser Arbeit sprengen würde, grundsätzlich werden die in einem Iterationsschritt aufgebauten, also all möglichen Baumstrukturen, nach vier Kriterien bewertet, sortiert und gefiltert. In jedem Iterationsschritt werden die am besten bewerteten Bäume beibehalten, am Ende der beste aus allen Iterationen ausgewählt (vgl. Cristea, Postolache und Pistol, 2006).

No of *edus*	No of sentences of this length	No of generated *edts* per sentence
1-3	25	1-4
4-5	9	5-28
6	1	42

Tabelle 2: Evaluierung der Discourse Structure (vgl. Cristea, Postolache und Pistol, 2006)

Die Kosten des vorliegenden Konzepts können sehr einfach aus Tabelle 4 abgelesen werden, da sofort ersichtlich ist, dass die Anzahl der Berechnungen aufgrund

der kettenförmigen Auswertung mit der Text und der Satzlänge fast exponentiell mit ansteigen kann. Da das Konzept für die automatische Textzusammenfassung verwendet wird, sind nur Precision, Recall und F-Measure im Vergleich zu der händischen und der automatischen Zusammenfassung von MS-Word angeführt und es können an dieser Stelle keine genauen Ergebnisse aufgeführt werden. Abschließend kann jedoch sagen, dass das Discourse Structure Modell sehr viel versprechende, wenn auch kostenintensive, Ergebnisse liefert und durchaus gut in der Lage ist, Koreferenzen und cue Phrasen in die Satzbewertung mit einzubeziehen (vgl. Cristea, Postolache und Pistol, 2006).

3.1.7 Support Vector Regression

Ouyang, Li und Li (2007) stellen in ihrer Arbeit *Developing Learning Strategies for Topic-based Summarization* einen Ansatz vor, der herkömmliche Text-Zusammenfassungssysteme mit einer automatischen Feature Gewichtung erweitert. Der Fokus liegt demnach auf verbesserten maschinellem Lernen und die Satzbewertung wird mittels Support Vector Regression (SVR), einem Derivat von Support Vector Machines approximiert.

Jeder Satz wird anhand von drei themenabhängigen und vier themenunabhängigen Features gewichtet. Die ersteren drei sind das *Word Matching Feature*, welches Wörtern, die in der Themenbeschreibung vorhanden sind, ein Gewicht verleiht, das *Semantic Matching Feature*, das für jedes Wort mittels WordNet die semantische Ähnlichkeit zu den Wörtern in der Themenbeschreibung berechnet und das *Name Entity Matching Feature*, das die Anzahl der Übereinstimmungen im Satz und der Beschreibung von Personen, Organisationen, Orten etc. mit einbezieht. Letztere vier sind das *Document Centroid Feature*, das die tf-idf Bewertung im ganzen Datensatz angibt, das *Named Entity Number Feature*, das die Anzahl an Named Entities im Satz angibt, das *Stop Word Penalty Feature*, das die Anzahl an Stoppwörtern im Satz enthält und das *Sentence Position Feature*, das die Satzposition im Dokument berücksichtigt, wobei früher auftretende Sätze ein höheres Gewicht erhalten (vgl. Ouyang, Li und Li, 2007).

Als Trainingsdaten werden Dokumente der DUC 2005 und DUC 2006 Sammlung, welche auch eine manuelle Zusammenfassung beinhalten, herangezogen. Jedem Uni-Gram beziehungsweise Bi-Gramm im Satz werden dabei je nach Auftreten in den händischen Zusammenfassungen Gewichte verliehen und deren Gewicht in den einzelnen Sätzen aufsummiert. Für die Berechnung der Auftrittswahrscheinlichkeit von Termen in den manuell getätigten Zusammenfassungen werden zwei Strategien, die *maximum* und die *average* Strategie, verfolgt. Bei der *maximum* Strategie werden die Wahrscheinlichkeiten für die Terme für jede einzelne Textzusammenfassung, bei der average Strategie die Wahrscheinlichkeit für

das Auftreten eines Terms in allen Zusammenfassungen (vgl. Ouyang, Li und Li, 2007).

Bei den meisten Ansätzen werden die einzelnen Feature, wie bereits erwähnt, aus linearen Kombinationen der Gewichte berechnet, in diesem Fall wird die Regressionsfunktion angelernt, indem die Struktur Risiko Funktion minimiert wird. Anschließend wird das Gewicht eines Satzes mit dem jeweiligen Wert der Regressionsfunktion multipliziert und die Sätze nach deren normierten Endgewicht sortiert. Da hierbei bei mehreren gleichen Termen in verscheidenden Sätzen die Redundanz sehr hoch sein kann, wird iterativ jeweils der beste Satz ausgewählt und der nächst bessere auf der Kandidatenliste hinzugefügt, sofern das Ähnlichkeitsmaß zu den bereits ausgewählten Sätzen nicht zu hoch ist. Die Auswertung der Ergebnisse erfolgte mittels dem ROUGE-2 Algorithmus mit einem Konfidenzintervall von 95 Prozent und zeigte, dass die maschinellen Zusammenfassungen oftmals eine höhere Bewertung erhielten als die von Hand getätigten Zusammenfassungen. Tabelle 3 verdeutlicht den Einfluss der oben genannten Satz Features und zeigt, dass eine Kombination aller dieser Features die besten Resultate liefert (vgl. Ouyang, Li und Li, 2007).

$f_{centroid}$	f_{word}	$f_{position}$	$f_{stopword}$	$f_{entity} + f_{entityno}$	$f_{wordnet}$	Average Rouge-2 and CI
√						0.06030 (0.05711, 0.06354)
	√					0.06280 (0.05981, 0.06563)
	√	√				0.06407 (0.06117, 0.06698)
√	√					0.07056 (0.06731, 0.07376)
√	√	√				0.07088 (0.06753, 0.07404)
√	√		√			0.07286 (0.06944, 0.07612)
√	√	√	√			0.07467 (0.07109, 0.07812)
√	√	√	√	√		0.07509 (0.07150, 0.07857)
√	√	√	√	√	√	**0.07556 (0.07201, 0.07912)**

Tabelle 3: Evaluierung der SVR Textzusammenfassung (vgl. Ouyang, Li und Li, 2007)

3.1.8 Random-Walk Termgewichtung

Ein Ansatz zur Optimierung der Gewichtung einzelner Terme der zur Textklassifikation genutzt wird, wurde von den Autoren Hassan, Mihalcea und Banea (2007) vorgestellt und berücksichtigt neben den Häufigkeiten der Terme auch den Kontext, in dem der Term auftritt. Ursprünglich wurde der Random-Walk Algorithmus dazu erfunden, anhand von Bewertungen und Empfehlungen ein Page Ranking im Web durchzuführen. Dabei werden einzelnen Seiten die Gewichte über die Anzahl der Verlinkungen auf diese und das Gewicht der verlinkenden Seiten verliehen. Die tatsächliche Abwandlung des PageRank Algorithmus die bei den obigen Autoren Verwendung findet nennt sich TextRank, wobei das Prinzip identisch bleibt. Es wird ein

Graph für das Dokument eingeführt, in dem jeder Term auf andere so genannte co-occurrences zeigt und die Gewichte auf diese Art und Weise „weitervererbt“ werden können.

$$S'(V_a) = \frac{(1-d)}{|N|} + \sum_{V_b \in In(V_a)} C * \frac{d_{E_{V_b,V_a}} * S(V_b)}{|Out(V_b)|}$$

Abbildung 5: Random-Walk Gewichtsberechnung (vgl. Hassan, Mihalcea und Banea, 2007)

Das Gewicht eines Knotens errechnet sich über die Formel in Abbildung 5, wobei d_{EVb} das, über die höchst gewichtete Kante normierte, Kantengewicht darstellt, $S(V_b)$ das Gewicht des Knotens der mit dem aktuellen über diese Kante verbunden ist und $Out(V_b)$ die vom aktuellen Knoten ausgehenden Gewichte beschreibt. N ist die Anzahl aller Knoten, d die Dämpfungskonstante und C die Skalierungskonstante (vgl. Hassan, Mihalcea und Banea, 2007).

Zu Beginn wird der Text tokenisiert und es werden häufige Gebrauchswörter anhand einer vorab definierten Liste eliminiert. Danach werden die Termgewichte und anhand obiger Formel die Random-Walk Gewichte für jeden Term errechnet. Ein so genannter co-occurrence Scanner sucht nun für jeden der Terme, die sich innerhalb einer definierten Umgebung, der window size, befinden, alle Terme, die eine Relation zu diesem aufweisen. Nachfolgend wird aus den gewonnen Informationen ein Graph erzeugt ausgewertet (vgl. Hassan, Mihalcea und Banea, 2007).

Abschließend wurden sowohl die Termgewichte als auch die Random-Walk Gewichte in Vektoren gespeichert und die Textklassifikation anhand eines naiven Bayes-Klassifikators, dem Rocchio Algorithmus sowie Support Vector Machines durchgeführt. Als Datensätze für die Evaluierung der Konzepte dienten Sammlungen aus WebKB, LingSpam und diversen Newsgroups. Die Ergebnisse waren beim Einsatz aller obig genannten Klassifikationsverfahren durchwegs besser als bei einem alleinigen Einsatz von Termgewichtungen, wobei im Detail die Fehlerrate der Klassifikation um bis zu 84 Prozent gesenkt werden konnte (vgl. Hassan, Mihalcea und Banea, 2007).

3.1.9 Häufige Sequenzen

Ein Ansatz zur Termextraktion anhand von häufigen Sequenzen wird von Ledeneva, Gelbukh und Garciá-Hernández (2008) in der Arbeit *Terms Derived from Frequent Sequences for Extractive Text Summarization* beschrieben. Das vorgestellte Konzept basiert auf statistischen Methoden und arbeitet vollkommen Unabhängig von Sprache und Kategorie. Die Grundlage für nachfolgende Überlegungen sind Wort N-Gramme, die den Autoren nach einerseits bei wiederholtem Auftreten im Dokument wichtig sind, aber abgesehen davon auch spezifisch sein müssen, da jene sonst

anders ausgedrückt werden würden. Einzelne Wörter erfüllen diese Bedingung nicht. Zusätzlich können N-Gramme auch von längeren N-Grammen beinhaltet werden, wobei in diesem Fall das längere N-Gramm vorzuziehen ist.

Darüber hinaus benennen Ledeneva, Gelbukh und Garciá-Hernández (2008) jene Sequenzen, die nicht von anderen häufigen N-Grammen, den *frequent sequences* FS, beinhaltet werden als *Maximal Frequent Sequences*, kurz MFS. Je nach definiertem Schwellenwert, also der mindestens vorausgesetzten Häufigkeit, können bei gleichem Text verschieden lange N-Gramme ermittelt werden. Der Algorithmus verwendet nur N-Gramme, die MFS sind, da diese wie bereits erwähnt einerseits wichtige Information in sich tragen und andererseits die Kosten der Auswertung von nicht maximalen Sequenzen den erreichbaren Nutzen übersteigen würde. Darüber hinaus können durch Zerlegung der MFS in die einzelnen Sub-N-Gramme die FS rekonstruiert werden. Die einzelnen durchgeführten Schritte sind die Auswahl der Terme nach bestimmten Kriterien, die anschließende Gewichtung, die Berechnung eines Satzgewichtes durch verschiedene Gewichtungen von Features und letztendlich die Satzauswahl für die endgültige Textzusammenfassung (vgl. Ledeneva, Gelbukh und Garciá-Hernández, 2008).

Als Testdatensatz fungierten dabei 567 News Artikel aus der DUC Kollektion, die nun der Text Analysis Conference (siehe auch TAC, 2010) angehört, die verschiedenen Kategorien zuzuordnen sind. In besagter Kollektion befinden sich zu jedem Artikel zwei von Experten erstellte Zusammenfassungen die zur abschließenden Evaluierung herangezogen werden(vgl. Ledeneva, Gelbukh und Garciá-Hernández, 2008).

Um als Term ausgewählt zu werden muss jener, je nach definiertem Schwellenwert n, mindestens n mal im Dokument vorkommen. Jene Terme werden nun mit deren Häufigkeit gewichtet und die Satzgewichte dementsprechend gesetzt. Nun wird der Schwellenwert für die Auswahl der Sätze sukzessive nach oben gesetzt, bis die ausgewählten Sätze in Summe ungefähr 100 Wörter beinhalten. Alternativ werden vor der Termauswahl die Stoppwörter ausgeklammert, wobei die Ergebnisse dieser einfachen Methodik mit Stoppwörtern einen Recall von etwa 0.43 und ohne 0.44 hervorbrachten. Dieser Wert kann geringfügig verbessert werden, indem alle N-Gramme, die ein Teil von den kalkulierten MFS sind, in Gewichtung der Sätze miteinbezogen werden. Es ist anzumerken, dass bei der Satzauswahl, neben den besten Sätzen (bis 100 Wörter erreicht sind), zusätzlich die ersten Sätze bei der Zusammenfassung hinzugefügt werden, da laut den Autoren Ledeneva, Gelbukh und Garciá-Hernández (2008) bei Zeitungsartikeln und wissenschaftlichen Publikationen die ersten Sätze die für das Textverständnis wichtigsten sind.

3.2 Automatische Fragengenerierung

In diesem Teil der Arbeit werden die bereits existierenden Ansätze zur voll- und semiautomatischen Aufgabenerstellung vorgestellt. Zusätzlich werden diese Verfahren evaluiert, um in weiterer Folge daraus die wichtigsten Konzepte entwickeln zu können.

Laut den Autoren Rus, Cai und Graesser (2007) werden die bisherigen Ansätze zur automatischen Fragengenerierung in drei wesentliche Arten unterschieden. Die erste Art von Systemen ist die *query/question reformulation*, wobei als Input eine Frage dient und daraus wiederum eine Frage oder eine Query erzeugt wird. Von *pseudo question generation* wird obigen Autoren nach dann gesprochen, wenn die generierten Fragen Lückentexte oder Multiple Choice Fragen sind und die Fragen nicht neu formuliert sondern in der ursprünglichen Form im Text sind, wobei ein oder mehrere Wörter ausgeblendet sind. Die dritte Art von Systemen, die eine automatische Fragengenerierung implementieren, sind die so genannten *limited question generation* Systeme. Bei diesen werden die generierten Sätze unmittelbar aus den Inputsätzen erzeugt und nach gewissen Regeln umgeformt. Ein Aussagesatz wird also beispielsweise in einen Fragesatz transformiert.

3.2.1 Corpus Word Frequency Data

Der Autor Coniam (1997) hat in seiner Arbeit *A Preliminary Inquiry Into Using Corpus Word Frequency Data in the Automatic Generation of English Language Cloze Tests* einen der ersten korpusbasierenden Ansätze vorgestellt, bei dem Wörtern Tags zugewiesen und Worthäufigkeiten ausgewertet werden, um daraus automatisiert auf drei verschiedene Arten Multiple Choice Fragen zu generieren. Unter einer Multiple Choice Frage ist hierbei ein Satz mit Lücken und jeweils fünf Antwortmöglichkeiten, also einer richtigen Antwort und vier Distraktoren zu verstehen.

Der Text wurde mittels dem Automatic Grammatical Tagging System annotiert und als erster Ansatzpunkt jedes *n-te* (zum Beispiel jedes sechste) Wort des annotierten Korpus durch eine Lücke ersetzt, wobei darauf zu achten ist, dass jene einer passenden Wortklasse, wie etwa Hauptwörtern, und nicht schlecht austauschbaren Wortklassen wie Artikeln oder dergleichen, angehören. In weiterer Folge wurde eine vom Bank of England Corpus abgeleitete Wortliste mit den 158.000 häufigsten Wörtern beziehungsweise deren häufigsten Wortformen erstellt. Für die ausgeklammerten *n-ten* Wörter werden nun in jener Wortliste die Häufigkeiten in Abhängigkeit der Wortklasse ermittelt (vgl. Coniam, 1997).

Zu den ausgewählten *test items* werden nun aus den alternativen Wörtern aus jener Wortliste, Wörter gleichen Typus mit ähnlicher Wahrscheinlichkeit ausgewählt, welche ihrerseits als Distraktoren fungieren. Bei jenen Distraktoren wird zusätzlich

darauf geachtet, dass die Groß - Kleinschreibung erhalten bleibt, Artikel davor angepasst werden und dass bei ähnlicher Häufigkeit mit einem vorigem *test item* die Häufigkeit erhöht wird, um gleichartige oder identische Distraktoren zu vermeiden (vgl. Coniam, 1997).

Die alternativen Verfahren zur Ermittlung der *test items* über jedes *n-te* Wort sind die Berechnung über die einzelnen *Worthäufigkeiten*, wobei jeweils die häufigsten Wörter mit adäquaten Wortklassen ausgesucht werden und die Auswahl, die nur auf *Wortklassen* basiert. Die restliche Vorgehensweise bleibt ident. Die Evaluierung der auf diese drei Arten erzeugten Tests verdeutlicht Tabelle 4 (vgl. Coniam, 1997).

Nth word based test items				Frequency based test items				Word class based test items			
Cloze 1		Cloze 2		Cloze 1		Cloze 2		Cloze 1		Cloze 2	
No. of items	Acc. items	No. of items	Acc. items	No. of items	Acc. items	No. of items	Acc. items	No. of items	Acc. items	No. of items	Acc. items
33	15	26	10	19	13	27	12	22	11	48	28
(45%)		(38%)		(68%)		(44%)		(50%)		(58%)	

Tabelle 4: Evaluierung der Ansätze von Coniam (vgl. Coniam, 1997)

Die Auswertung von Tabelle 4 zeigt, dass die drei Ansätze relativ verschiedene Ergebnisse hinsichtlich der Anzahl der errechneten Lücken aufweisen und die auf die jeweils *n-ten* Wörter aufbauende Strategie in Bezug auf die als angemessen bewerteten Fragen im Vergleich am schlechtesten abschneidet. Der Wortklassenansatz und der Häufigkeitsansatz liefern demnach in etwas bessere Ergebnisse und können im Schnitt immerhin 50 Prozent als passend deklarierte *test items* vorweisen (vgl. Coniam, 1997).

3.2.2 Exercises in Adaptive Hypermedia Learning Systems

Fischer und Steinmetz (2000) stellen in der Arbeit *Automatic Creation of Exercises in Adaptive Hypermedia Learning Systems*, basierend auf dem Multi Book System (siehe auch Multibook, 1999), das es dem Nutzer erlaubt, adaptiv individuelle Informationen unter Berücksichtigung der Bedürfnisse und Vorlieben zu erhalten, einen Ansatz vor, der anhand von Mustern und Konzepten automatisiert Aufgaben aus einem Lernsystem erstellen kann. Bevor der Ansatz zur automatischen Aufgabenerstellung dargelegt wird, werden nachfolgend die wichtigsten Konzepte des Multi Book Systems vorgestellt.

Das Multi Book System, das dem IEEE LTSA (Learning Technology System Architecture) Standard unterliegt, arbeitet auf zwei verschiedenen Ebenen, dem *ConceptSpace* und dem *MediaBrickSpace*. Der *ConceptSpace* nutzt dabei Ontologien in Bezug auf Schlüsselwörter, um einen Rahmen für die dargebotenen Inhalte

zu erstellen. Zusätzlich beinhaltet diese Ebene eine Sammlung an semantischen Relationen, die zwischen den einzelnen Einträgen bestehen und in Tabelle 5 verdeutlicht werden. Der *MediaBrickSpace* hat die Aufgabe, diesen Rahmen mit wirklicher Information zu füllen und beinhaltet rhetorische sowie didaktische Relationen, die zwischen den einzelnen *MediaBricks* vorherrschen. Der Vorteil dieses Modells ist die einfache Erweiterbarkeit und Veränderbarkeit sowohl der Inhalte als auch des Rahmens, in dem die Inhalte dynamisch eingebettet werden (vgl. Fischer und Steinmetz, 2000).

Name of relation	Explanation
Superconcept	A node is a superconcept of another node.
AEPart	For all instances of a node there exists a subnode
EEpartOf	There exists a subnode for an instance of a super-node
(Inverse)Procedure	A node contains a (an inverse) procedure with regard to another node.
Follows/Precedes	A node follows/precedes another node (ordering in a document)
Formalize/Is FormalizedBy	A node formalizes / is formalized by another node
ProblemSolution	A node points to a problem-node which is connected to a solution-node.
Partition	Subnodes partition a domain, for example images are partitioned as b/w, gray, and color.
Cost	The cost of another node
Uses	A node uses another node
Application	A node is an application of another node.
Instance	A node is an instance of another node

Tabelle 5: Relationen im ConceptSpace des MultiBook Systems (vgl. Fischer und Steinmetz, 2000)

Die automatische Aufgabenerstellung beruht dabei vollständig auf den Relationen (siehe Tabelle 5) im *ConceptSpace* und lässt zwei wesentliche Fragestellungen zu. Der erste Fragentypus implementiert Multiple Choice Fragen der Form „*Which are the parts of <name of the concept>?*“, wobei die richtigen Antworten anhand der Auswertung der *uses* Relationen des erfragten Konzepts bestimmt werden. Darüber hinaus werden Distraktoren ermittelt, indem Teile eines *superconcepts* vom vorliegenden *superconcept* ermittelt werden, wobei *instance* Relationen bei Bedarf übersprungen werden können, und von jenem wiederum die *uses* Relationen ausgewertet werden. Dies kann solange für verschiedene hierarchische Ebenen durchgeführt werden, bis zufrieden stellende Distraktoren ermittelt sind. Diese Vorgehensweise erzwingt eine semantische Nähe der Distraktoren zur Frage, da durch die

superconcepts gewährleistet ist, dass zwei voneinander vollkommen verschiedene Konzepte niemals in Relation zueinander stehen (vgl. Fischer und Steinmetz, 2000).

Der zweite Fragentypus hat die Form „*What are the applications of <name of the concept>?*", wobei für die richtigen Antworten die *application* Relation des erfragten Konzepts ausgewertet wird. Für die Distraktorenwahl ist ebenso das obige Prinzip anwendbar, indem die *uses* gegen die *application* Relation ausgetauscht wird. Bei beiden Arten von Fragen werden die richtigen Antworten und die Distraktoren in der Reihenfolge randomisiert und auf Basis der Antworten des Benutzers können dessen Schwachstellen durch Einblendung der sowohl falsch als auch richtig verwendeten Konzepte aufgezeigt werden (vgl. Fischer und Steinmetz, 2000).

Das vorgestellte Konzept beruht auf der Auswertung von Relationen, die vorab exakt definiert werden müssen und hat demnach den Vorteil dass dadurch bereits semantische Relationen gegeben sind. Der große Nachteil ist jedoch, dass der Vorgang abseits des Multi Book Systems dadurch nur semi-automatisch erfolgen kann, wen die Implementierung von derartigen Mustern und Konzepten fehlt.

3.2.3 Computer-Aided Generation of Multiple-Choice Tests

Die Autoren Mitkov und Ha (2003) stellen einen Ansatz zur automatischen Multiple Choice Test Generierung vor, der im wesentlichen auf einfachen Transformationsregeln, einem Shallow Parser, automatischer Term Extraktion, Wortbedeutungsunterscheidung, einem Korpus und dem WordNet Tool (siehe auch Kapitel 5.2) basiert.

Im ersten Schritt gilt es dabei die so genannten *anchors*, also für die Textkategorie spezifische Terme zu bestimmen. Zur Termextraktion, vorübergehend werden nur Hauptwörter und Hauptwortphrasen ermittelt, wird dabei der FDG Shallow Parser von Tapanainen und Järvinen (1997) benutzt und jene nach Häufigkeit geordnet. Darüber hinaus wird ein Schwellenwert festgelegt, wobei in weiterer Folge nur Hauptwörter, die eine größere Häufigkeit als der Schwellenwert besitzen, genutzt. Zusätzlich werden nun Phrasen, die ein ausgewähltes Hauptwort, einen so genannten *key term*, beinhalten und dem Schema von Justeson und Katz (1996), wonach reguläre Ausdrücke [AN]+N sowie [AN]*NP[AN]*N gesucht werden, genügen, als *terms* bezeichnet. Alternativ werden mit Hilfe von WordNet Phrasen für *key terms* gesucht und aufgrund von oftmals zahlreichen Bedeutungen eines Wortes mit dem Kontext im vorliegenden Textkorpus abgeglichen (vgl. Mitkov und Ha, 2003).

Zusätzlich müssen für jeden dieser Terme passende Distraktoren ermittelt werden, die einerseits semantisch gesehen sehr nahe an der Antwort liegen, andererseits jedoch keinerlei Hilfestellung in Bezug auf die richtige Antwort geben sollen. Als kleines Beispiel führen die Autoren Mitkov und Ha (2003) den vorab extrahierten

Term „syntax“ und die Distraktoren „pragmatics“ und „semantic“ an. Grundsätzlich werden Distraktoren mittels WordNet gesucht, indem Hypernyme, Hyponyme und Koordinaten des Frageterms ermittelt werden. Dabei werden die Ergebnisse von WordNet wieder mit dem Korpus abgeglichen, um eine größtmögliche semantische Nähe zu gewährleisten. Besonders zu beachten ist hierbei, dass jenes Konzept nur auf die *key terms* angewandt werden kann, da das WordNet für Phrasen sehr wenig Resultate liefert.

Die tatsächliche Aufgabenstellung wird nach Berechnung obig genannter Terme durch Transformation von deklarativen Sätzen erreicht, indem demnach einfache Konzepte zur Wortvertauschung angewandt werden. Dabei muss ein in Frage kommender Satz entweder zumindest die Struktur Subjekt-Verb oder Subjekt-Verb-Objekt aufweisen. Ein SVO Aussagesatz wird dabei beispielsweise in einen „Which Hypernym-Verb-Objekt“ oder in einen „What does Subjekt-Verb“ Fragesatz transformiert. Um eine grammatikalische Korrektheit zu gewährleisten müssen die Distraktoren, sofern das zu erfragende Wort in Pluralform vorkommt, ebenso im Plural verwendet werden (vgl. Mitkov und Ha, 2003).

Die Evaluierung des Ansatzes zeigte, dass etwa 57 Prozent der generierten Aufgaben als mögliche Fragestellungen geeignet waren, wobei wiederum sechs Prozent davon ohne jegliche Nachbearbeitung verwendet werden konnten. In Bezug auf die Zeit, die für die automatische Aufgabenerstellung gebraucht wurde, ergab sich eine Rechenzeit von neun Stunden für 300 Fragen. Vergleichsweise dazu würde eine manuelle Aufgabenerstellung in etwa die fast vierfache Zeit in Anspruch nehmen. Zur rechnerischen Auswertung der Ergebnisse wurden die Probanden der Tests in zwei Gruppen aufgeteilt, nämlich in die Hälfte der besseren und der schlechteren Resultate. Den Schwierigkeitsgrad der Aufgaben, die Trennschärfe und die Distraktorenqualität der automatisch erstellten im Vergleich zu den manuell erstellten Aufgaben verdeutlicht die Tabelle 6.

	item difficulty			item discriminating power		usefulness of distractors			
	avg item difficulty	too easy	Too difficult	average discriminating power	negative discriminating power	poor	not useful	Total	avg difference
computer-aided	0.75	3	0	0.4	1	6	3	65	1.92
manual	0.59	1	0	0.25	2	10	2	33	1.18

Tabelle 6: Vergleich der Effizienz von automatisch und manuell erstellten Aufgaben (vgl. Mitkov und Ha, 2003)

Aus obiger Tabelle ist sofort ersichtlich, dass die Schwierigkeit der generierten Aufgabenstellungen höher als jener der manuell Erstellten ist. Außerdem sagt die Trennschärfe aus, dass der Leistungsunterschied zwischen schlechteren und besseren Probanden bei computergenerierten Fragen größer ist, wonach die Distrakto-

ren gut gewählt sind. Der durchschnittliche Unterschied jener zwei Gruppen bei der Punktevergabe untermauert diese These und gibt Aufschluss über die Sinnhaftigkeit des Gesamtkonzepts von Mitkov und Ha (2003).

3.2.4 Fragengenerierung im REAP System

Das REAP System wurde von Brown, Frishkoff und Eskenazi (2005) entwickelt, um den Nutzern Fragestellungen geben zu können, die deren Wissensstand angepasst sind. Dazu werden als Grundlage für die Tests aus dem Web Inhalte nach gewissen Kriterien ausgewählt. Jene Texte müssen einen gewissen Anteil an vom jeweiligen Nutzer bekanntem Vokabular beinhalten, wobei diese Inhalte aus circa 95 Prozent bekannten Termen bestehen sollen. Nachdem der Nutzer einen Text gelesen hat, werden die restlichen fünf Prozent der bis dato unbekannten Vokabeln in Form von Multiple Choice Fragen und Zuordnungsfragen abgeprüft. Anhand dieser Ergebnisse wird anschließend das jeweilige User Modell adaptiert.

Bei der Auswahl von Vokabular wird der Fokus auf die konzeptionelle Bedeutung dieses gelegt, da es den Autoren Brown, Frishkoff und Eskenazi (2005) darum geht, dass die Nutzer des REAP Systems das gelernte Vokabular nicht rein nach der Wortbedeutung, sondern im Kontext anwenden können. Um die konzeptionelle Bedeutung eines Wortes zu extrahieren wird das WordNet, das in Kapitel 5.2 noch genauer erläutert wird, eingesetzt und daraus unter anderem Definitionen und Phrasen einzelner Wörter extrahiert.

Es kommen bei der Fragengenerierung letztendlich sechs Typen, nämlich Fragen, die auf Definitionen, Synonyme, Antonyme, Hypernyme und Hyponyme abzielen, sowie Lückentexte zum Einsatz. Auf die genaueren Bedeutungen der obigen Begriffe wird nicht näher eingegangen, da dies nicht Gegenstand der Arbeit ist. Da Wörter in WordNet oftmals etliche Bedeutungen haben, wird vordergründig der POS Tag eines Wortes im Text ausgewertet und in weiterer Folge die häufigste Wortbedeutung aus WordNet gesucht. Selbiges Prinzip kommt auch bei Suche nach Hyper- und Hyponymen zum Einsatz (vgl. Brown, Frishkoff und Eskenazi, 2005).

Bei Multiple Choice Fragen werden immer vier Antwortmöglichkeiten angegeben, wobei die Anzahl der Distraktoren und der richtigen Lösungen stets variiert wird. Als Distraktoren kommen bei Synonym-, Hyponym-, Hyperonym- und Antonymfragen nach Möglichkeit nur Wörter mit ähnlich häufiger Verwendung im allgemeinen Sprachgebrauch, die anhand der Kilgarriff Datenbank (Kilgarriff, 2010) ermittelt wird, und selber Wortklasse zum Einsatz. Bei Definitionsfragen und Lückentexten sollen die Distraktoren zusätzlich zu dieser Methode auf dem erfragten Wort basieren, also zu diesem eine Relation in WordNet aufweisen. Alternativ können jedoch auch Distraktoren anhand von semantischer Ähnlichkeit zu anderen Wörtern im selben Text ausgewählt werden (vgl. Brown, Frishkoff und Eskenazi, 2005).

Als Evaluierung für die generierten Fragen kamen 156 seltene Wörter zum Einsatz, die in vorhergehenden Studien zur Beurteilung von englischsprachigen Erwachsenen herangezogen wurden. Zu erwähnen ist hierbei, dass Hypernyme und Hyponyme ausgeklammert wurden, da dieses Konzept nicht auf Adjektive anwendbar ist, jene jedoch einen größeren Anteil in besagter Wortliste einnehmen. Letztendlich konnten, bedingt durch die Resultate aus WordNet, für 50 Prozent der Wörter aus der List für alle vier übrigen Konzepte Fragen generiert werden. In Tabelle 7 sind die genauen Ergebnisse für die vier evaluierten Konzepte ersichtlich (vgl. Brown, Frishkoff und Eskenazi, 2005).

Question type	Percentage of Questions Generated
Definition Question	91%
Synonym Question	80%
Antonym Question	60%
Cloze Question	60%

Tabelle 7: Prozentsatz der generierten Fragen nach Typ (vgl. Brown, Frishkoff und Eskenazi, 2005)

Als Vergleichswert für die Qualität der generierten Fragen wurden von unabhängigen Personen manuell Fragen mit verschiedenem Schwierigkeitsgrad erstellt. Auf das genaue Testsetup wird aufgrund mangelnder Relevanz für die weitere Betrachtung nicht näher eingegangen. In Summe waren, je nach Typ, 52 bis 64 Prozent der Fragen nutzbar. Die Korrelation zwischen den automatisch generierten und den händisch erstellten Fragen wies bei allen Typen mindestens 70 Prozent, bei Fragen zu Synonymen sogar 90 Prozent auf (vgl. Brown, Frishkoff und Eskenazi, 2005).

3.2.5 Real-time multiple-choice

Ein auf maschinellem Lernen basierender Ansatz findet sich in der Arbeit *A real-time multiple-choice question generation for language testing: a preliminary study* von den Autoren Hoshino und Nakagawa (2005), die den Naiven Bayes-Klassifikator und den k-Nearest-Neighbor Algorithmus verwenden, um aus Nachrichtenartikeln von Online Portalen Multiple-Choice Fragen zu generieren. Dabei wird der Fokus auf die auszuwählenden Fragewörter und deren Lücken im Text, nicht aber auf die Umformulierung von Fragesätzen oder die Distraktorenwahl gelegt.

Da dieser Ansatz durch maschinelles Lernen charakterisiert ist, muss zu Beginn eine Trainingsmenge, in diesem Fall eine Sammlung aus dem TOEIC (Test of English for International Communication) fill-in-the-blank Bestand, erstellt werden um die Klassifikatoren trainieren zu können. Als Ausgangspunkt dienen 100 Fragen, deren leere Position an alle möglichen Positionen gewechselt wird, wobei die originalen Lücken mit *true*, die anderen Möglichkeiten mit *false* markiert sind. Natürlich sind

alle mit *true*, aber auch einige mit *false* markierten Positionen potentielle Möglichkeiten für eine Fragestellung. Anhand des naiven Bayes Klassifikators werden nun genaue jene als *false* markierten Lücken die den Wert von 0,5 übertreffen nachträglich als *true* markiert (vgl. Hoshino und Nakagawa, 2005).

Im nächsten Schritt wurden nun News Artikel auf dieselbe Art und Weise getaggt und anhand der Trainingsmenge, also deren im Endeffekt 113 sinnvollen Positionen, klassifiziert. Nach Ermittlung der leeren Stellen in den vorliegenden Artikeln wurden nachfolgend Distraktoren aus dem Text gesucht, die jedoch ohne zusätzliche semantische und morphologische Analysen ausgewählt wurden. Als Alternative kam der KNN Algorithmus zum Einsatz, dessen Einsatz von Hoshino und Nakagawa (2005) jedoch nicht genauer beschrieben wird.

Die Evaluierung dieses Systems wurde durch Kategorisierung der ausgewerteten Lücken in die Klassen E, die gut geeigneten, D, die möglichen aber schwierigen und NG, die, etwa durch Satzzeichen, nicht möglichen Positionen unterteilt. Die Ergebnisse und Vergleiche der zwei genannten Klassifikationsverfahren finden sich in Tabelle 8.

	NB				KNN				I
	blanks	E(%)	D(%)	NG(%)	blanks	E(%)	D(%)	NG(%)	blanks
Article1	69	44(63.8)	21(30.4)	4(5.8)	33	20(60.6)	11(33.3)	2(6.1)	18
Article2	22	5(22.7)	3(13.6)	14(63.6)	8	5(62.5)	3(37.5)	0(0.0)	0
Article3	38	21(55.3)	15(39.5)	2(5.3)	18	12(66.7)	5(27.8)	1(5.6)	8
Article4	19	10(52.6)	9(47.4)	0(0.0)	9	7(77.8)	2(22.2)	0(0.0)	3
Article5	28	18(64.3)	10(35.7)	0(0.0)	14	10(71.4)	4(28.6)	0(0.0)	6
Article6	26	17(65.4)	8(30.8)	1(3.8)	11	6(54.5)	5(45.5)	0(0.0)	4
Article7	18	9(50.0)	5(27.8)	4(22.2)	6	3(50.0)	3(50.0)	0(0.0)	3
Article8	24	14(58.3)	9(37.5)	1(4.2)	5	3(60.0)	2(40.0)	0(0.0)	5
Article9	20	16(80.0)	4(20.0)	0(0.0)	6	2(33.3)	4(66.7)	0(0.0)	4
Article10	30	18(60.0)	12(40.0)	0(0.0)	14	11(78.6)	3(21.4)	0(0.0)	6
	294	172(58.5)	96(32.7)	26(8.8)	124	79(63.7)	42(33.9)	3(2.4)	57

Tabelle 8: Vergleich der NB und KNN Klassifikation (vgl. Hoshino und Nakagawa, 2005)

Aus obiger Tabelle wird ersichtlich, dass der k-Nearest-Neighbor Algorithmus weniger und meistens auch bessere Lücken klassifiziert. Die Autoren Hoshino und Nakagawa (2005) merken darüber hinaus auch an, dass der naive Bayes Klassifikator vermehrt Zustandsverben als Lücken wählt und dazu tendiert, oftmals die gleichen Wörter auszuwählen. Die Spalte I beschreibt die Anzahl der von beiden Klassifikationsverfahren gleich berechneten Positionen und zeigt, dass im Schnitt etwa nur die Hälfte der Lücken übereinstimmen. Ein generelles Problem beider Verfahren ist jenes, dass die Lücken unabhängig voneinander ausgewählt werden und dadurch bei Lücken in unmittelbarer Umgebung Schwierigkeiten in Bezug auf das Textverständnis verursachen. Dahingehend geben die Autoren auch an, dass es zielführend ist, für weitere Überlegungen semantische Informationen zu berücksichtigen und die Trainingsmengen genauer zu untersuchen. Der große Vorteil dieses

Konzepts ist jedoch die Unabhängigkeit von der Sprache, der Textkategorie und der Semantik, wodurch schnell ohne Zusatzwissen Ergebnisse erreicht werden.

3.2.6 FAST

Die Autoren der Arbeit *FAST – An Automatic Generation System for Grammar Tests,* Chen, Liou und Chang (2006) entwickelten ein System, das eine semi-automatische Aufgabenerstellung von grammatikalischen Fragen mittels manuell erstellten Mustern ermöglicht. Als Beispiel für ein Muster wird der Satz „The weather tends to improve in May“ angeführt, wobei hier das Verb „improve“ extrahiert und für die Distraktorenerzeugung herangezogen werden. Anhand dieses Musters werden im Web authentische Sätze gesucht. Abbildung 6 zeigt die aus diesem Muster letztendlich generierte Multiple-Choice Frage, Abbildung 7 die Muster in Bezug auf die Bildung der Distraktoren.

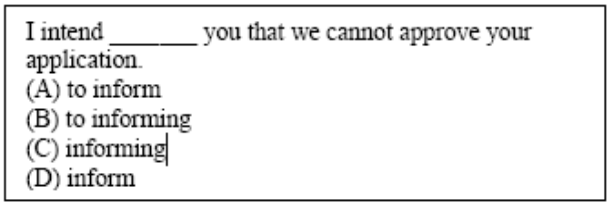


Abbildung 6: Beispiel einer Multiple Choice Aufgabe (vgl. Chen, Liou und Chang, 2006)

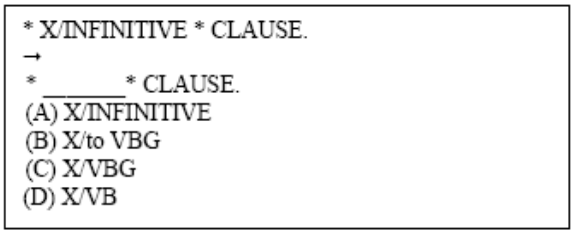


Abbildung 7: Muster für die Distraktorenwahl (vgl. Chen, Liou und Chang, 2006)

Für die Aufgabenerstellung werden Informationen aus Webseiten ohne HTML Tags ausgelesen, auf diesen Texten wird POS Tagging, Lemmatisierung und Chunking ausgeführt und letztendlich wird versucht die händisch erstellten Muster auf die einzelnen Sätze anzuwenden. Ein Muster sieht im konkreten folgendermaßen aus: „{* finish X/Gerund *}. Ein möglicher Satz muss demnach das Verb „finish“ und ein darauf folgendes Gerundium beinhalten. Für die Distraktorenkonstruktion werden dabei für jedes Muster wiederum eigene Muster erzeugt. Für das Verb „writing“ mit dem Muster {* VBD VBG *} hätten die Distraktoren die Form {$0 VB} für „write“, {$0 VBD} für „wrote“ und {$0 VBN} für „written“, wobei $0 für eine Änderung des Pivot Wortes steht (vgl. Chen, Liou und Chang, 2006).

Zusätzlich zur Multiple-Choice Aufgabenerstellung wurden auch Aufgaben zur so genannten error detection kreiert. Dabei wurden einerseits die selben Muster wie oben angeführt verwendet, andererseits aber auch Chunks gleichen Types zu einer Chunk Phrase verschmolzen („the/B-NP nickname/I-NP“ wird zu „the nickname/NP“) und dazu Distraktoren erzeugt. In einem Satz werden dann richtige Phrasen gegen grammatikalisch falsche Phrasen ausgetauscht und sind vom Probanden zu erkennen (vgl. Chen, Liou und Chang, 2006).

Für die Evaluierung dieses Konzepts wurden 69 strukturelle Muster für neun grammatikalische Kategorien entworfen und Artikel aus Wikipedia und der VOA (Voice of American) Sammlung entnommen. Die generierten Fragen wurden von sieben Professoren und Studenten bewertet und 77 Prozent der Multiple-Choice Fragen sowie 80 Prozent der error detection Sätze für brauchbar befunden (vgl. Chen, Liou und Chang, 2006).

3.2.7 Questions About Facts

Die Autoren Rus, Cai und Graesser (2007) stellen in ihrer Arbeit *Experiments on Generating Questions About Facts* einen Ansatz vor, dem eine erweitertes XML Schema zu Grunde liegt und einen Interpreter mit kontext-sensitiven Primitiven implementiert, die linguistische Kriterien bewerten können. Dieser Interpreter ist demnach in der Lage, Regeln auf neue Daten anzuwenden und daraus Fragen zu generieren. Unter einer Regel versteht man die Kombination aus einem Muster und einem Template, das eine lexikalische, semantische und syntaktische Struktur verkörpert. Ein Muster wiederum stellt jene Konditionen dar, die ein Template auslösen.

Das auf Mustern basierte System besteht dabei aus zwei Kernkomponenten, ein Teil zur Verwaltung und Umsetzung der Muster und Templates, den so genannten Kategorien, und einem zum Erzeugen von Fragen, die sich an jenen Kategorien orientieren. Aus dem Trennen vom Inhalt und dem Generationsmodul resultiert ein Erweiterbarkeit und Austauschbarkeit. Zusätzlich lassen sich den Autoren Rus, Cai und Graesser (2007) nach auch Regeln, die Prioritäten von Mustern betreffen, einsetzen. Außerdem ist dadurch gewährleistet, dass anhand von wenigen Mustern aus dem Kontext eine große Anzahl von Mustern erzeugt werden kann.

Das für diesen Ansatz entwickelte erweiterte XML Schema trägt den Namen QG-ML (Question Generation Markup Language) und basiert auf dem AIML Schema (Artificial Intelligence Markup Language), das für die automatische Sprachgenerierung, im speziellen bei Chat Bots die natürlichsprachliche Antworten geben, eingesetzt wird. Wie bereits erwähnt sind die Kategorien, bestehend aus Mustern und Templates, eines der Kernkonzepte dieser Sprache. Im Grunde genommen besteht die syntaktische Struktur von QG-ML aus einem syntaktischen Baum, lexikalischen und syntaktischen Beschränkungen, diversen Variablen und Funktionsaufrufen so-

wie Regeln für Prioritäten von Mustern und Templates. (vgl. Rus, Cai und Graesser, 2007).

Ein Blattknoten des syntaktischen Baumes ist entweder ein Wort, eine Variable oder ein Funktionsaufruf. Wenn ein Muster jene Variable matcht, so wird der gesamte Teilbaum in das zugehörige Template eingesetzt. Wenn im Knoten ein Funktionsaufruf steht, so wird dieser in weiterer Folge durch den Rückgabewert ersetzt, in einem Template wird dadurch kontextabhängig vorgegangen. Ein Beispiel für eine Kategorie, also ein Muster und ein Template, in QG-ML ist in Abbildung 8 zu finden. Der Tag NP (siehe auch Kapitel 5.1.1) kennzeichnet dabei eine Hauptwortphrase, die Variable _np_ steht für den Inhalt dieser (vgl. Rus, Cai und Graesser, 2007).

```
<category>
<pattern>
<NP>_np_</NP>
</pattern>
<template>
What can you say about _np_?
</template>
</category>
```

Abbildung 8: Beispiel einer QG-ML Kategorie (vgl. Rus, Cai und Graesser, 2007)

Die Evaluierung des Konzeptes wurde mit 200 Fragen und Antworten aus der TREC-8 QA Sammlung (TREC, 2010) durchgeführt. Die TREC-8 QA Sammlung besteht aus so genannten Faktenfragen, sprich die Antworten sind sehr kurz und prägnant gehalten. Einige kleine Beispiele für derartige Fragen sind in Abbildung 9 aufgezeigt. Anschließend wurden Kategorien erzeugt, die genau dieselben Fragen wie jene der TREC-8 QA Sammlung zu den angegeben Antworten generieren. Die Precision für die generierten Fragen lag in einem Bereich von 0.52 bis 0.64 (vgl. Rus, Cai und Graesser, 2007).

Type	Question
WHO	Who is the voice of Miss Piggy?
WHAT	What does the Peugeot company manufacture?
WHERE	Where is Microsoft's corporate headquarters located?
WHEN	When did Nixon die?
HOW	How many people live in the Falklands?
OTHER	Name the first private citizen to fly in space.

Abbildung 9: Beispiele aus TREC QA (vgl. Rus, Cai und Graesser, 2007)

3.2.8 Limited-Choice and Completion Test Creation

Der Autor Gütl (2008a) stellt in seiner Arbeit *Automatic Limited-Choice and Completion Test Creation, Assessment and Feedback in modern Learning Processes* eine

Vorgehensweise zur automatischen Erzeugung von Multiple-Choice Fragen und Lückentexten vor. Zur Ermittlung der zu erfragenden Terme werden Techniken zur automatischen Textzusammenfassung genutzt.

Ein Modul namens *Document Fetcher* ermöglicht dabei das Erfassen der Informationen und der *Document Filter* konvertiert die vorliegenden Daten in ein internes Dateiformat, das neben dem eigentlichen Inhalt auch Metainformationen beinhaltet. Zur Berechnung der relevantesten Terme werden folgende fünf statistischen Methoden angewandt: Es werden non jedem Wort die Worthäufigkeit (*word frequency method*) bestimmt, spezifische Phrasen gesucht (*cue phrase method*), Sätze nach deren Auftreten am Beginn oder Ende bevorzugt (*location method*), Wörter mit dem Titel auf Ähnlichkeit geprüft (*title method*) und Benutzereingaben bei der Aufgabenerstellung berücksichtigt (*query method*) (vgl. Gütl, 2008a).

Bevor die wichtigsten Terme gesucht werden wird eine automatische Zusammenfassung erzeugt, indem der Text mittels GATE tokenisiert, die Satzgrenzen ermittelt, POS Tagging und morphologische Analysen durchgeführt und Stoppwörter entfernt werden. Zusätzlich werden Listen verwendet, die vordefinierte themenspezifische Wörter beinhalten und ebenso zur Extraktion von Schlüsselwörtern genutzt werden können (vgl. Gütl, 2008a).

Der *Test Word Extractor* erstellt basierend auf obigen Informationen liefert eine gewisse Anzahl an Kanditatenwörtern, die gemeinsam mit dem erzeugten Abstract zur Aufgabenstellung, welche das *Exercise Creator* Modul bewerkstelligt, herangezogen werden. Bei der Erstellung von Lückentexten werden im Abstract die ausgewählten Wörter ausgeblendet, bei Multiple-Choice werden zusätzlich Antonyme und Synonyme ermittelt, um als Distraktoren zu fungieren (vgl. Gütl, 2008a).

3.3 Automatisches Assessment

In diesem Kapitel soll ein Einblick über die existierenden Ansätze in Bezug auf automatisches Assessment beziehungsweise, auf maschinellem Lernen und NLP Methoden basierende, Konzepte zur Bewertung von Texten nach verschiedensten Kriterien gegeben werden. Dabei werden einerseits bewährte Ansätze beschrieben, andererseits auch neuartige, noch nicht ganz ausgereifte Ansätze vorgestellt.

3.3.1 Project Essay Grade

Der Autor Hearst (2000) schreibt, dass der erste Ansatz zur automatischen Beurteilung, der im Project Essay Grade Anwendung findet, auf Ellis Page zurückgeht. Jener hat versucht, anhand von sprachlichen Eigenschaften eines Textes die Qualität

einer Arbeit zu beurteilen, indem die automatische Beurteilung eines Aufsatzes mittels einer Kombination von Gewichtungsfaktoren bestmöglich an die Bewertungen von Lehrern angepasst wurde. Die Art und Weise der Beurteilung war dabei aufgrund der beschränkten Möglichkeiten in den 60er Jahren sehr oberflächlich, zum Einsatz kamen dabei die Kriterien der durchschnittlichen Wortlänge, des Aufsatzumfangs, der Anzahl der Beistriche und Präpositionen sowie der Anzahl der verwendeten eher ungebräuchlichen Wörter (vgl. Hearst, 2000).

Die Probleme dieses ersten Versuches waren vielfältig: Die Korrelation der Benotung mit jener der Lehrer betrug 78 Prozent, die Schwachstellen waren jedoch die indirekte Beurteilung und die leichte Überlistbarkeit des Systems, da allein die Aufsatzlänge unabhängig von der Qualität enorme Vorteile bringen konnte. Inhaltliche Eigenschaften und der Stil wurden demnach vollkommen ausgeklammert (vgl. Hearst, 2000).

3.3.2 Intelligent Essay Assessor

Der Intelligent Essay Assessor baut auf Latent Semantic Analysis (LSA) auf und bietet laut Hearst (2000) die Möglichkeit die semantische Übereinstimmung zweier Dokumente unabhängig des verwendeten Vokabulars zu beurteilen. Dabei werden allen Wörtern und Phrasen mittels maschinellen Lernens Gleichungen zugeordnet, welche die Bedeutungen untereinander ausdrücken und jene in einen höheren semantischen Raum projizieren. Dabei werden diese in die so genannte co-occurrence Matrix übergeführt. Die Ähnlichkeit von Wörtern wird demnach über deren relative Positionen in diesem Raum festgelegt. Ein Vorteil dieses Verfahrens ist die Unabhängigkeit von Sprache, da keine Muster und Regeln von Hand erzeugt werden müssen.

Als Trainingsdaten kommen je nach Fachgebiet etwa Lehrbücher zum Einsatz, da davon ausgegangen wird, dass diese den optimalen Inhalt kurz und prägnant widerspiegeln. Die wesentlichen Bewertungsfaktoren sind beim Intelligent Essay Assessor der Inhalt, der anhand der Trainingsmenge klassifiziert wird, der Stil, wobei die Kohärenz bewertet wird und die Syntax, also die Ermittlung von Rechtschreibfehlern. Zusätzlich sind in das System Mechanismen zur Plagiatsüberprüfung und zur Validitätskontrolle eingebaut (vgl. Hearst, 2000).

Der Vorteil von LSA ist das von Hearst (2000) genannte *vicarious human scoring*, wonach die Bewertungskriterien nicht von Hand erzeugt werden müssen, sondern sich implizit aus den Trainingsdaten ergeben. Darüber hinaus werden sowohl irrelevante als auch redundante Sätze in die Bewertung miteinbezogen. Die Nachteile dieses Systems sind die große benötigte Menge an zu bewertenden Aufsätzen, die mit 200 beziffert wird und die fehlende Möglichkeit, reine Fakten zu beur-

teilen. Die Korrelation mit der Beurteilung von Experten liegt bei etwa 85 Prozent (vgl. Hearst, 2000).

3.3.3 AEGIS

Der *Automatic Exercise Generator based on the Intelligence of Students* ist ein System von Mine, Suganuma und Shoudai (2000), dass basierend auf dem Kenntnisstand von Studenten gezielt Aufgaben verschiedener Schwierigkeitsgrade erstellen können. Zusätzlich bietet das System die Möglichkeit gegebene Antworten zu evaluieren und geeignetes Feedback zu geben. Unterstützt werden Multiple-Choice Fragen, Lückentexte und Fehler-Korrektur Fragen. Alle jene Fragen können durch Ersetzen der so genannten *hidden region* durch eine andere Phrase erreicht werden. Darüber hinaus wird die *question region* als Bereich beziehungsweise Satz in dem die *hidden region* liegt bezeichnet.

Das Dokument wird nun mit den Tags *Question*, einem umschließenden Tag der *question region*, *Del*, dem umschließenden Tag der *hidden region*, und *Label*, dem Namen, sowie den Tags, die eine Relation einer hidden region mit einem Teil des Dokumentes angeben, gekennzeichnet. Außerdem werden für den Schwierigkeitsgrad die Tags *Level*, ein Wert der diesen beeinflusst, *Group,* welcher benötigt wird um Relationen zwischen *hidden regions* zu halten, und *Ref*, dessen Aufgabe es ist, Relationen einer *hidden region* mit anderen Textteilen zu kennzeichnen, eingeführt (vgl. Mine, Suganuma und Shoudai, 2000).

Das AEGIS System benötigt im Hintergrund drei Datenbanken, die excercise DB, die User Profile DB und die Level Management DB. Darüber hinaus besteht das System aus drei Modulen, einem, welches die Fragen generiert (EG), einem, das die Fragen auswertet (AE) und letztendlich jenem Modul für den Schwierigkeitsgrad (LM). Die Zusammenhänge der einzelnen Systemkomponenten sind in Abbildung 10 ersichtlich, die wesentlichen Kernkomponenten werden nachfolgend kurz beschrieben (vgl. Mine, Suganuma und Shoudai, 2000).

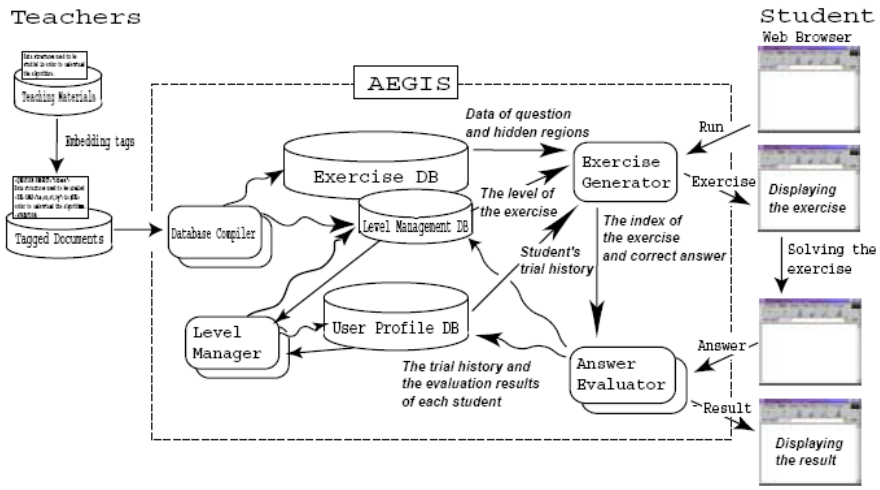


Abbildung 10: AEGIS Systemarchitektur (vgl. Mine, Suganuma und Shoudai, 2000)

Das EG Modul sucht in der Datenbank, welche die Aufgaben enthält, in Abhängigkeit der Daten, die den Schwierigkeitsgrad der jeweiligen Frage und den Wissensstand des Nutzer zu jenem Gebiet betreffen, die am besten zutreffende *hidden region* zum gewünschten Thema. Genauer gesagt wird eine Multiple-Choice Frage ausgesucht, wenn der Wissensstand des Nutzers sehr nahe am niedrigsten deklarierten Schwierigkeitsgrad der Frage liegt. Eine Fehler-Korrektur Frage wird dann bevorzugt, wenn der Wissensstand eher näher am der höchsten festgelegten Schwierigkeitsgrad liegt. Der Aufbau der letztendlich erzeugten Fragen und die Auswahl der Distraktoren wurden bereits mehrfach in Kapitel 3.2 behandelt, weshalb hier nicht mehr näher darauf eingegangen wird (vgl. Mine, Suganuma und Shoudai, 2000).

Das AE Modul erhält in weiterer Folge vom EG Modul den Index der *hidden region*, den kalkulierten Fragetyp und die richtige Antwort auf die Frage. Es wertet anhand dieser Informationen die Frage aus und speichert die Ergebnisse in der Datenbank mit den Benutzerprofilen. Das LM Modul, updatet nun in Abhängigkeit vom Fortschritt des Wissensstandes eines Nutzers die Initialwerte für den Schwierigkeitsgrad, welche manuell von Lehrern erzeugt wurden, nach den folgenden Schemata. Wird eine Frage, die einen geringeren Schwierigkeitsgrad hat als der Wissensstand des Nutzers eingestuft ist, falsch beantwortet, so wird der Frage sofort ein höherer Grad zugeordnet. Selbiges gilt für den analogen umgekehrten Fall, woraus eine Abstufung resultiert. Nebenbei wird in jedem Schritt in Abhängigkeit der Anzahl der richtig beantworteten schweren Fragen der Level des Nutzers adaptiert und vice versa. Eine Evaluierung anhand von jeweils hundert Fragen in den zehn möglichen Schwierigkeitsgraden zeigte, dass wenn Studenten ihrem Wissensstand entsprechende Fragen beantworten, eine Standardabweichung von 0.78 bei leichteren, 0.87 bei mittleren und 0.96 bei schwierigen Fragen erreicht wird (vgl. Mine, Suganuma und Shoudai, 2002).

3.3.4 E-Rater

Der E-Rater wurde von Burstein, Kukich, Wolff, Chi und Chodorow entwickelt und nutzt neben NLP Methoden auch statistische Methoden um linguistische Kriterien zu extrahieren. Dabei werden vordergründig das Einhalten der Thematik, die kohärente und gut organisierte Argumentation sowie die syntaktische Struktur und der Wortgebrauch beziehungsweise die Qualität des eingesetzten Vokabulars bewertet (vgl. Valenti, Neri und Cucchiarelli, 2003).

Die Implementierung des Systems besteht aus fünf voneinander unabhängigen Modulen. Drei davon ermitteln jene Eigenschaften eines Textes, welche zur Bewertung herangezogen werden. Dies sind die syntaktischen Variationen, der Aufbau von Argumentationskonzepten und der Einsatz von Vokabular. Ein viertes Modul gewichtet jene Eigenschaften, ein letztes errechnet abschließend die tatsächliche Bewertung (vgl. Valenti, Neri und Cucchiarelli, 2003).

Die Evaluierung des E-Raters, der mittels 270 von Experten bewerteten Aufsätzen trainiert wurde, bescheinigt nach der Beurteilung von 750000 Aufsätzen eine Übereinstimmung der Benotung im Vergleich zur händischen Benotung von 97 Prozent (vgl. Valenti, Neri und Cucchiarelli, 2003).

3.3.5 Syntactically Enhanced LSA

Die Autoren Kanejiya, Kumar und Prasad (2003) erweitern in ihrer Arbeit *Automatic Evaluation of Students' Answers using Syntactically Enhanced LSA* das in Kapitel 3.3.2 vorgestellte LSA Modell, bei dem das Kosinus Ähnlichkeitsmaß der Projektionen in den LSA Raum zweier beliebiger Texte berechnet wird, um die Berücksichtigung von den jeweiligen Wortumgebungen, die sich aus den syntaktischen Regeln der Wortklasse ergeben. Es wird demnach der Nachteil von LSA, dass Textfragmente als so genannter *bag-of-words* gesehen werden und sowohl die Wortreihenfolge als auch syntaktische Informationen vernachlässigt werden, ausgeglichen.

Beim Syntactically Enhanced LSA (SELSA) wird jedes Wort mit dem POS Tag des vorhergehenden Wortes erweitert. Je nach POS Tag des vorangegangenen Wortes kann also die syntaktisch-semantische Bedeutung eines Wortes variieren. Anstelle der herkömmlichen co-occurrence Matrix wird der Text in eine Matrix transformiert, in deren Reihen alle möglichen Wort – POS Tag Kombinationen gehalten werden und deren Spalten den Dokumenten entspricht. Anstatt nur Dokumente in die Spalten einzutragen lässt sich dieses Problem auch auf einzelne Texte reduzieren, indem in den Spalten jeweils Paragraphen oder Sätze eingetragen werden. Sobald diese Matrix aufgestellt ist, wird diese approximiert, indem eine gewisse Anzahl n der größten Werte beibehalten und die restlichen Werte auf Null gesetzt wer-

den. Daraus ergibt sich die neue n x n Repräsentation, die *syntactically enhanced latent semantic space* genannt wird (vgl. Kanejiya, Kumar und Prasad, 2003).

Im Endeffekt wird bei diesem Modell für jedes Dokument ein Vektor aufgestellt werden, dessen Werte Häufigkeiten für einzelne Wörter, die in Kombination mit verschiedenen *previous tags* auftreten, gewichtet mit der Entropie derer, darstellen. Das Ähnlichkeitsmaß zweier Dokumente, oder eben auch Absätze oder Sätze, können demnach einfach über den Winkel, den zwei Vektoren aufspannen, berechnet werden (vgl. Kanejiya, Kumar und Prasad, 2003).

Für die Evaluierung dieses Konzeptes wurde ein Trainingskorpus aus zwei Büchern über Computer und zehn spezifischen Artikeln aufgestellt. Das Training erfolgte mit 5596 daraus erzeugten Dokumenten, in diesem Fall Paragraphen, und es wurden, nach einigen Vorverarbeitungsschritten, 9194 Wörter ermittelt. Aus jener Anzahl an Dokumenten wurden für die Tests die 200 bis 400 Besten in die genannte Matrixschreibweise transformiert. Anschließend wurden Studenten gebeten acht Fragen über Themen aus der Trainingsmenge zu beantworten. Experten beurteilten dann von Hand die in Summe 192 Antworten der Studenten und erhielten zusätzlich Referenzantworten, die zuvor erstellt wurden. Die Ergebnisse einer herkömmlichen LSA konnten von SELSA nicht übertroffen werden, da die syntaktische Komponente offensichtlich wenig bis keinen Einfluss auf das Ähnlichkeitsmaß zu Referenzantworten haben. Positiv anzumerken ist jedoch, dass SELSA wesentlich toleranter gegenüber Schwellenwerten ist und ein größeres Spektrum dieser ermöglicht (vgl. Kanejiya, Kumar und Prasad, 2003).

3.3.6 BETSY

Das Bayesian Essay Test Scoring sYstem (BETSY) bedient sich zwei verschiedenen Arten des Bayes Modell, dem *Multivariate Bernoulli Model* und dem *Multinominal Model*. Bei Ersterem wird die Wahrscheinlichkeit für das Auftreten eines Features durch die Anzahl der Aufsätze einer Kategorie abgeschätzt, in dem das Feature beinhaltet ist, wohingegen beim zweiten Model die Wahrscheinlichkeit des Auftretens eines Features als Produkt der Wahrscheinlichkeiten aller Features in dem Dokument berechnet wird. Der Fokus des Bernoulli Modells liegt also darin, zu bestimmen ob ein Feature in einem Text auftritt, beim Multinominalen Modell wird das mehrfache Auftreten von Features in einem Dokument berücksichtigt (vgl. Dikli, 2006).

Zu Begin werden, wie üblich, Stemming, Stoppwort Elimination und die Feature Auswahl, welche durch Reduktion der Entropie geschieht, durchgeführt. Unter einem Feature versteht man hierbei spezifische, aber auch in Abhängigkeit von der Worthäufigkeit, Satzlänge, Satzzeichenanzahl etc. ermittelte Wörter und Phrasen. Anhand von 1000 Textdokumenten wird das BETSY trainiert und sukzessive Wörter

angelernt, ungebräuchliche Wörter beziehungsweise unübliche Wort Chunks entfernt und das Trainingsset bewertet sowie falsch kategorisierte Texte ausgeschnitten. Ein zu klassifizierender Text wird dabei in zwei Gruppen, *pass* und *fail*, und in eine vierstufige Skala eingeordnet, was einer Notenvergabe entspricht, und der Nutzer kann anhand vielfältiger Einstellmöglichkeiten in diesen Prozess eingreifen. Die konstatierte Genauigkeit der Klassifikation mit BETSY liegt bei 80 Prozent (vgl. Dikli, 2006).

3.3.7 Essay Scoring Using KNN

Die Autoren Bin, Jun, Jian-Min und Qiao-Ming (2008) stellen in der Arbeit *Automated Essay Scoring Using the KNN Algorithm* einen Ansatz zur automatischen Bewertung von Aufsätzen vor, der auf dem K-Nearest Neighbor Algorithmus zur Textklassifikation beruht. Dazu wird einerseits das Dokument in ein Vector Space Model transformiert, andererseits werden die Termgewichte und die zugehörige invertierte Dokument Frequenz berechnet.

Um eine Dimensionsreduktion des Vektorraums zu erreichen werden die folgenden zwei wesentlichen Methoden angewandt. Einerseits werden Phrasen und Wörter mit einer Häufigkeit unter einem gewissen Schwellenwert nicht weiter berücksichtigt, andererseits wird versucht, anhand von Wahrscheinlichkeiten vorherzusagen, welche Kategorien bestimmte Features beinhalten. Dazu wird ein großer Korpus von Trainingsdokumenten aufgestellt und berechnet, welche Auftrittswahrscheinlichkeit Kategorien, einzelne Features und Dokumente einer Kategorie, die bestimmte Features enthalten, haben und welche Dokumente diese nicht enthalten (vgl. Bin, Jun, Jian-Min und Qiao-Ming, 2008).

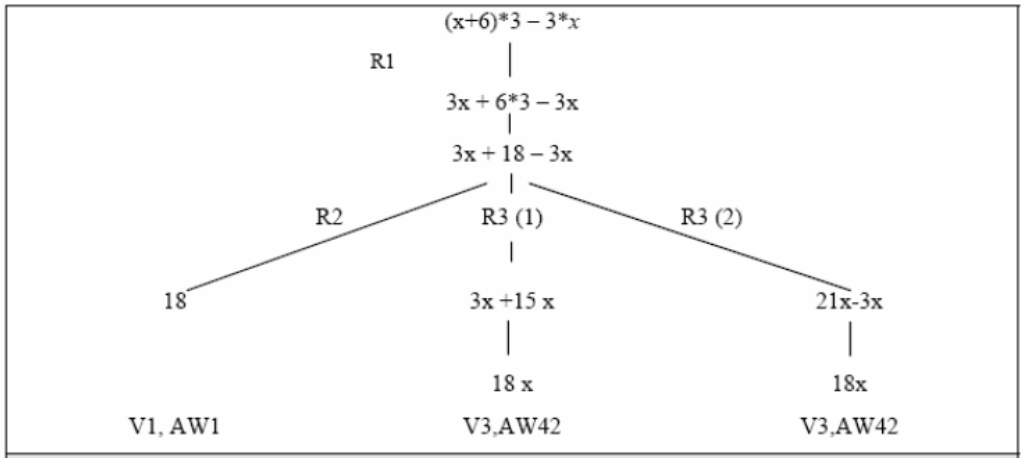
Zu Beginn der Klassifikation werden die Stoppwörter eliminiert und mehrere Schwellenwerte eingeführt, um zu seltene beziehungsweise zu häufige, wenig aussagekräftige Wörter auszuklammern. Aus den transformierten Vektormodellen mit den tf-idf Gewichten für die Trainingsdokumente werden nun die nächsten Nachbarn für jedes Klassifikationsdokument gesucht, indem das Kosinus Ähnlichkeitsmaß berechnet wird. Die besten Resultate, eine Genauigkeit von 76 Prozent, werden für die drei und fünf nächsten Nachbarn erreicht. Ein wesentlicher Faktor für dieses Ergebnis war unter anderem die Elimination der häufigsten Features einer Kategorie (vgl. Bin, Jun, Jian-Min und Qiao-Ming, 2008).

3.3.8 Automatic Multi-criteria Assessment

Die Autoren Delozanne, Prévit, Grugeon und Chenevotot (2008) haben ein System Namens PépiGen entwickelt, das auf Pépite, einem früheren Tool der Autoren basiert. In diesem sind Muster für Modelle implementiert, die eine Menge von Aufga-

ben, Sichtweisen von Schülern in Bezug auf bekannte Lösungen und multidimensionales Assessment für jeden Lösungsansatz bieten. Dabei sind zur Auswertung der Open-Ended Antworten zwei wesentliche Designs notwendig. Es müssen eine Menge an Antworten für jeden möglichen Lösungsansatz zu einer Frage und für jeden Fragentypus der häufigste Lösungsansatz von Schülern nach empirischen Regeln aufgestellt werden.

Die Bewertung einer Antwort erfolgt nach 36 Kriterien in sechs Dimensionen, wobei die so genannten *personal features* ebenso mit einbezogen werden. Dabei werden sowohl der Einsatz von Algebra, die Verschiebungen eines Schemas in ein anderes als auch Berechnungen an sich mit allen anderen Nutzer Profilen verglichen und für die Bewertung herangezogen. Die sechs bei diesem Ansatz bewerteten Dimensionen sind die Validität, die Bedeutung von Buchstaben, algebraisches Schreiben, Querverbindungen zwischen verschiedenen Repräsentationen, die Art wie begründet wird und das numerische Schreiben (vgl. Delozanne, Prévit, Grugeon und Chenevotot, 2008).



Applied rules		
Correct rules	R1 : (A+B)C → AC+BC	R2 : AC+BC→ (A+B)C
Incorrect rules	R3 : A+BC→ (A+B)*C	Applied to : (1) 18 – 3x (2) 3x +18
Codes		
Validity	V1 : correct	V3 : incorrect
Algebraic Writing	AW1 : Correct use of rewriting rules	
	AW42 : The student gathers the terms in an algebraic sum.	

Abbildung 11: Baumstruktur des multi-criteria assessment (vgl. Delozanne, Prévit, Grugeon und Chenevotot, 2008)

Die erstellten Aufgaben an sich sind wie in Abbildung 11 in einer Art Baumstruktur gespeichert, wobei die einzelnen durchzuführenden Schritte jeweils einer Ebene in jener Struktur entsprechen und als Knoten dargestellt werden. Für jeden Lösungsschritt wird ein algebraisches oder numerisches Muster gespeichert. Zusätz-

lich werden für jeden Schritt ein oder mehrere gängige falsche Regeln und Muster abgeleitet. Anhand dieser Vorgehensweise lassen sich im Vergleich mit den Lösungen der Schüler, die auf die selbe Art und Weise aufgeschlüsselt werden, sofort Fehler erkennen, Schwachstellen aufzeigen und nützliches Feedback erzeugen (vgl. Delozanne, Prévit, Grugeon und Chenevotot, 2008).

3.3.9 e-Examiner

Der Autor Gütl (2008b) stellt in der Arbeit *Moving towards a Fully Automatic Knowledge Assessment Tool* das Tool e-Examiner, welches einen formativen Ansatz von Assessment umsetzt, vor. Es können damit sowohl automatisch Aufgaben erstellt werden, die gegebenen Antworten evaluiert als auch maschinelles Feedback gegeben werden. Die grundlegende Architektur des e-Examiner ist in Abbildung 12 dargestellt. Durch den Einsatz des IMS QTI Standard zeigt sich die Unabhängigkeit von spezifischen Systemen. Darüber hinaus sind die einzelnen Module der Implementierung vollkommen austauschbar und flexibel.

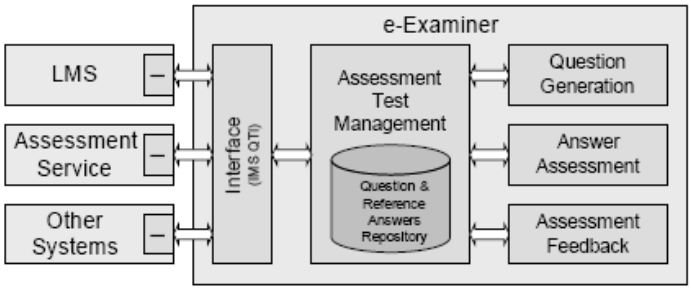


Abbildung 12: Architektur des e-Examiner (vgl. Gütl, 2008b)

Das für die Fragengenerierung zuständige Modul erstellt, basierend auf den wichtigsten Konzepten, Aufgaben und Referenzantworten, die in weiterer Folge für die Auswertung herangezogen werden. Es werden dabei offene Fragen der Form „Erklären sie das Konzept X“ erzeugt, bei Multiple-Choice Fragen werden Konzepte und Textpassagen im Umfeld des zu erfragenden Konzept extrahiert und als Distraktoren zur Verfügung gestellt. Die Fragen werden wie bereits erwähnt im IMS QTI Standard implementiert (vgl. Gütl, 2008b).

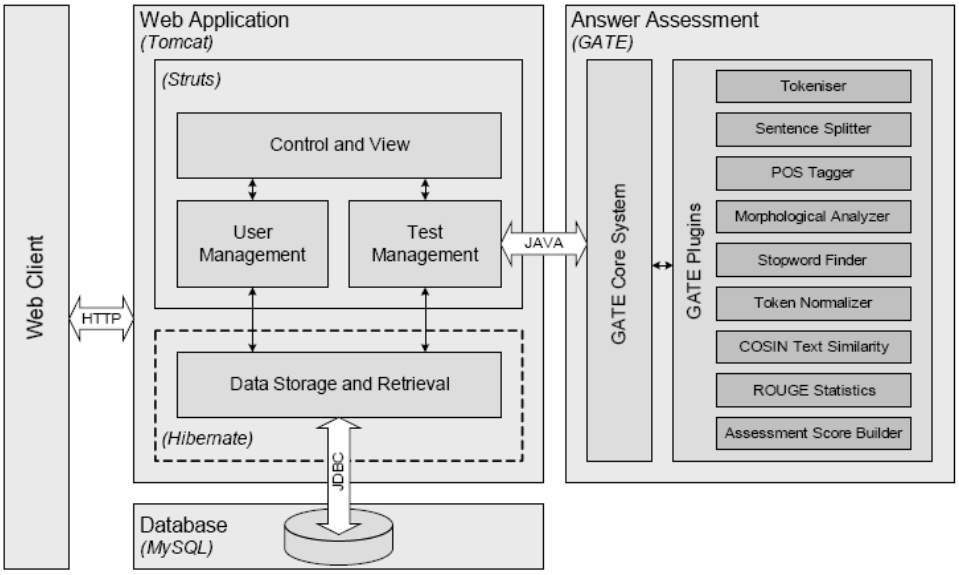


Abbildung 13: Assessment im e-Examiner (vgl. Gütl, 2008b)

Jenes Modul, das die automatische Beurteilung von Fragen umsetzt, ist in Abbildung 13 aufgezeigt, wobei sich zeigt, dass die Fragenauswertung beziehungsweise die nötigen vorhergehenden Schritte mit GATE Plugins bewerkstelligt werden. Der Text wird tokenisiert, Satzgrenzen detektiert, POS Tagging und morphologische Analysen durchgeführt, Stoppwörter eliminiert und die einzelnen Token werden normalisiert. Anschließend werden, basierend auf dem Vector Space Model, die Antwort sowie die Referenzantworten mit dem COSIN Text Similarity Plug-in auf Ähnlichkeit geprüft (vgl. Gütl, 2008b).

Interessant ist vor allem die darauf folgende Auswertung der Ergebnisse mittels dem ROUGE (Recall-Oriented Understudy for Gisting Evaluation) Statistics Plug-in. Es kommen dabei die folgenden ROUGE Metriken zum Einsatz: ROUGE-N ist ein Maß für die Übereinstimmung von Wort N-Grammen, ROUGE-L und ROUGE-W für die längste gemeinsame Subsequenz von Wörtern, ROUGE-S und ROUGE-SU für jene Bi-Gramme, zwischen welchen beliebige Lücken auftreten können. Anhand dieser Bewertungskriterien, obiger Ähnlichkeitsberechnung und einer Selbsteinschätzung der Lernenden kann der Lehrende gezielte Informationen über den Kenntnisstand der betreuten Lernenden erlangen. Ebenso können eine Auswertung in Bezug auf die gesamten Kursteilnehmer durchgeführt werden und auf diese Art und Weise Schwachstellen im Lehrsystem identifiziert werden (vgl. Gütl, 2008b).

Die Evaluierung des Systems in Bezug auf offene Fragen wurde anhand von Datensätzen des IICM (Institut für Informationssysteme und Computer Medien) der TU Graz und der technischen Fakultät der Al-Quds Universität durchgeführt. Insgesamt standen acht Fragen und jeweils 23 Referenzantworten von Studenten zur Verfügung, die von einem Experten mit null (schlecht) bis zehn Punkten (gut) bewer-

tet wurden. Anschließend wurden je Frage elf Antworten als Trainingsmenge herangezogen und die restlichen zwölf klassifiziert. Die Klassifikation erfolgt mittels dem ROUGE-1-P Maß, welches die Gesamtanzahl von Wörtern und die Precision einzelner Wörter angibt, mittels einer Kombination der Kosinus-Ähnlichkeit, Recall und Precision von einzelnen Wörtern und Bi-Grammen, ROUGE-l, ROUGE-S, ROUGE-SU und ROUGE-W sowie der Anzahl der Token, Recall und Precision der Wörter und dem ROUGE-L Recall. Alle drei genannten Setups wurden auf die Anzahl der absoluten Fehler und die Korrelation der Test- und Trainingsdaten untersucht beziehungsweise ausgewertet. Die Korrelation betrug stets zwischen 79 und 80 Prozent, die Anzahl der absoluten Fehler lag im schlechtesten Fall, dem zweiten Setup, bei drei, die wenigsten Fehler wurden mit dem dritten Setup erreicht (vgl. Gütl, 2008b).

3.4 Zusammenfassung

In diesem Kapitel wurden sowohl richtungweisende als auch sehr innovative Ansätze zu den Themen automatische Term- und Konzeptextraktion, automatische Aufgabenerstellung und automatisches Assessment vorgestellt. Es wurde aufgezeigt wie vielfältig die verschiedenen Konzepte, die dahingehend verfolgt werden, sind und welche Probleme diesen zu Grunde liegen.

Bezüglich der Extraktion von Schlüsselkonzepten ist besonders die Abhängigkeit von den Domänen hervorzuheben. Bei maschinellem Lernen etwa ist die Qualität der Klassifikation zu einem großen Teil durch die Trainingsmenge charakterisiert, bei anderen Ansätzen hängt die Gewichtung von Faktoren ab, die sehr spezifisch sind. Ein allgemein gültiges und fast immer als Grundlage für weitere Berechnungen eingesetztes Konzept ist jenes der Worthäufigkeit. Weiters zeigt sich, dass das WordNet sowie spezielle erweiterte Annotationen und Textstrukturen essentiell für die Term- und Konzeptextraktion sind.

In Bezug auf die automatische Aufgabenerstellung konnte aufgezeigt werden, dass diese Disziplin relativ wenig erforscht ist und noch keine wirklich zufrieden stellenden Ansätze existieren. Ein besonderer Aufholbedarf besteht bei der Formulierung von offenen Fragen, da fast alle Ansätze Fragen nach einem bestimmten Schema generieren oder dabei immer Textfragmente herangezogen werden. Relativ gute Ergebnisse werden hingegen bei Multiple-Choice Fragen erzielt, wobei die Distraktoren meist in WordNet ermittelt werden. Einige Verfahren, die vollkommen regelbasiert arbeiten, liefern zwar gute Resultate, sind aber immer nur in einem gewissen Kontext, wie dem Erlernen einer Sprache, einsetzbar.

Beim automatischen Assessment gibt es sehr viele gute Ansätze, wobei die wesentlichen Beurteilungsmerkmale auf semantischen und statistischen Analysen beruhen. Viele der eingesetzten Algorithmen benötigen als Vergleichsmenge Bewertungen von Experten, bei fast allen Ansätzen wird der Text in Vektoren oder Matrizen transformiert um Berechnungen durchführen zu können.

Über die Herstellung von Querverbindungen der einzelnen Konzepte soll im nächsten Kapitel eine optimale Lösung für das Problem der Entwicklung und des Designs von Konzepten zur automatischen Fragengenerierung erstellt werden. Jene wird in weiterer Folge die Grundlage für die diese Arbeit begleitende Implementierung darstellen.

4 *Anforderungen und Design*

Dieses Kapitel gibt einen Überblick darüber, welche Anforderungen ein umfassender Ansatz, welcher eine automatisierte Fragengenerierung umsetzt, erfüllen muss und wie ein mögliches konzeptionelles Design aussehen kann. Das in diesem Kapitel beschriebene Assessment System ist vor allem im Hinblick auf die Aufgabenstellung, dem Entwurf von Konzepten zur automatisierten Aufgabenerstellung beschrieben, einige Details wie Benutzerkonten und –datenbanken sowie eine Server-Client Architektur wurden wegen fehlendem Bezug zur direkten Aufgabenstellung außer Acht gelassen.

4.1 Anforderungen

Im Zuge dieser Arbeit soll eine modulbasierte Implementierung erstellt werden, welche ausgehend von einem Input Dokument eine vollautomatische Aufgabenerstellung umsetzt. Wie bereits aufgezeigt wurde ist die Verarbeitung von Texten immer sprachabhängig, der Fokus dieser Arbeit liegt deshalb auf englischsprachigen Texten, die Fragen sollen ebenso in englischer Sprache erzeugt werden.

Es sollen die wichtigsten Ansätze des theoretischen Teils dieser Arbeit berücksichtigt beziehungsweise kombiniert und erweitert werden. Die Anforderungen an ein System, das eine automatische Aufgabenerstellung umsetzt sind mannigfaltiger Natur, nachfolgend wird ein kurzer Überblick über die funktionellen und nicht funktionellen Anforderungen gegeben.

4.1.1 Allgemeine Lösung

Besonders hervorzuheben ist der Wunsch nach einer allgemeinen Lösung, da fast alle existierenden Ansätze, die Kapitel 3 vorgestellt wurden, dies nicht erfüllen. Ansätze, die sich vorwiegend maschinellem Lernen bedienen sind besonders themenspezifisch, da die Trainingsmenge die Resultate in Bezug auf die Klassifikation bestimmt und sind aufgrund dessen eher weniger geeignet.

4.1.2 Textuelle Vorverarbeitung

Die textuelle Vorverarbeitung beinhaltet die Unterstützung der gebräuchlichsten Dokumentenformate wie Adobe PDF, Microsoft Word, HTML, XML, SGML, RTF, ODT, E-Mail und Plain Text. Der Inhalt von Dokumenten soll in ein einheitliches

Format transformiert werden, welches in weiterer Folge die Schnittstelle für sämtliche Module darstellen soll.

4.1.3 Auswahl von Schlüsselkonzepten

Um eine automatische Fragengenerierung und Aufgabenerstellung ermöglichen zu können ist es erforderlich, die wichtigsten Wörter, Konzepte und Sätze eines Textes zu ermitteln. Dies kann wie bereits in den Kapiteln 3.2 und 3.1 gezeigt wurde auf vielfältige Art und Wiese erfolgen, die vom Autor vorgeschlagene Lösung wird im Laufe der Arbeit noch genauer erklärt und herausgearbeitet.

4.1.4 Fragetypen

Es sollen Open Ended, Single Choice, Multiple Choice und Fill In The Blank Fragen behandelt werden. Wie bereits in Kapitel 2.2.2 dargelegt sind die Anforderungen an eine Frage vom Einsatzgebiet abhängig. In dieser Arbeit liegt der Fokus auf dem Assessment von E-Learning Inhalten, weshalb die Anforderungen auf diese Problemstellung reduziert werden.

- **Open Ended Fragen**

Eine Open Ended Frage soll je nach erfragtem Konzept beziehungsweise in der Schlüsselphrase beinhalteten Annotationstypen nach einem gewissen Schema aufgebaut werden.

- **Single Choice Fragen**

Bei Single Choice Fragen soll dem Nutzer zu geeigneten Sätzen etwa durch Umformung eines Fragesatzes in einen Aussagesatz oder durch Vertauschung bedeutungstragender Wörter eine Ja/Nein Entscheidung geboten werden.

- **Multiple Choice Fragen**

Bei Multiple Choice Fragen sollen geeignete Distraktoren zu einer Phrase eines wichtigen Satzes ermittelt werden, die einerseits eine Ähnlichkeit zu der erfragten Phrase aufweisen, andererseits aber ohne Wissen auch nicht eindeutig als falsch zu erkennen sind.

- **Lückentexte**

Bei Lückentexten sollen gewisse bedeutungstragende Phrasen ausgeblendet werden und müssen vom Lernenden sinngemäß in die erzeugte Lücke eingesetzt werden.

4.1.5 Benutzerinteraktion

Der Nutzer soll die Möglichkeit haben aktiv in den Prozess der Schlüsselphrasen- und Satzauswahl haben. Darüber hinaus soll ein GUI entworfen werden, die es erlaubt diverse Settings zu testen um die Ergebnisse der erzeugten Aufgaben verbessern zu können. Zusätzlich soll die Möglichkeit bestehen sämtliche Zwischenergebnisse zu speichern und zu laden, damit der iterative Prozess bei Änderungen in einer Verarbeitungsstufe nicht immer von Beginn an gestartet werden muss.

Die Lernziele des Nutzers sollen durch geeignete Maßnahmen bestmöglich unterstützt werden. Durch eine voreingestellte Parametrisierung kann gewährleistet werden, dass der kontextuelle Einsatz in beliebigen Themenbereichen möglich ist.

4.1.6 Standardisierung

Die erzeugten Aufgaben sollen optimalerweise im QTI Standard gespeichert werden, um eine unproblematische Einbindung in ein E-Learning beziehungsweise E-Assessment System zu gewährleisten.

4.1.7 Fragenauswertung

Im Zuge der Aufgabenerstellung soll zu den extrahierten Schlüsselkonzepten eine Referenzantwort erstellt werden, welche in weiterer Folge dazu genutzt werden kann, die gegebenen Antworten zu auf deren Richtigkeit zu verifizieren.

4.1.8 Feedback

Auf dieselbe Art und Weise sollen dem User die bei einer gegebenen Antwort fehlenden Schlüsselterme beziehungsweise Textpassagen angezeigt werden können.

4.2 Konzeptionelles Design

In diesem Abschnitt wird eingangs in Abbildung 14 ein umfassendes konzeptionelles Design vorgestellt. Es ist ersichtlich, dass ein vollständiges Assessment System im Wesentlichen aus vier Teilen, dem Preprocessing, der Konzeptermittlung, dem Assessment und einer GUI, besteht. Anschließend werden kurz die darin beinhalteten Komponenten und deren Funktion beschrieben.

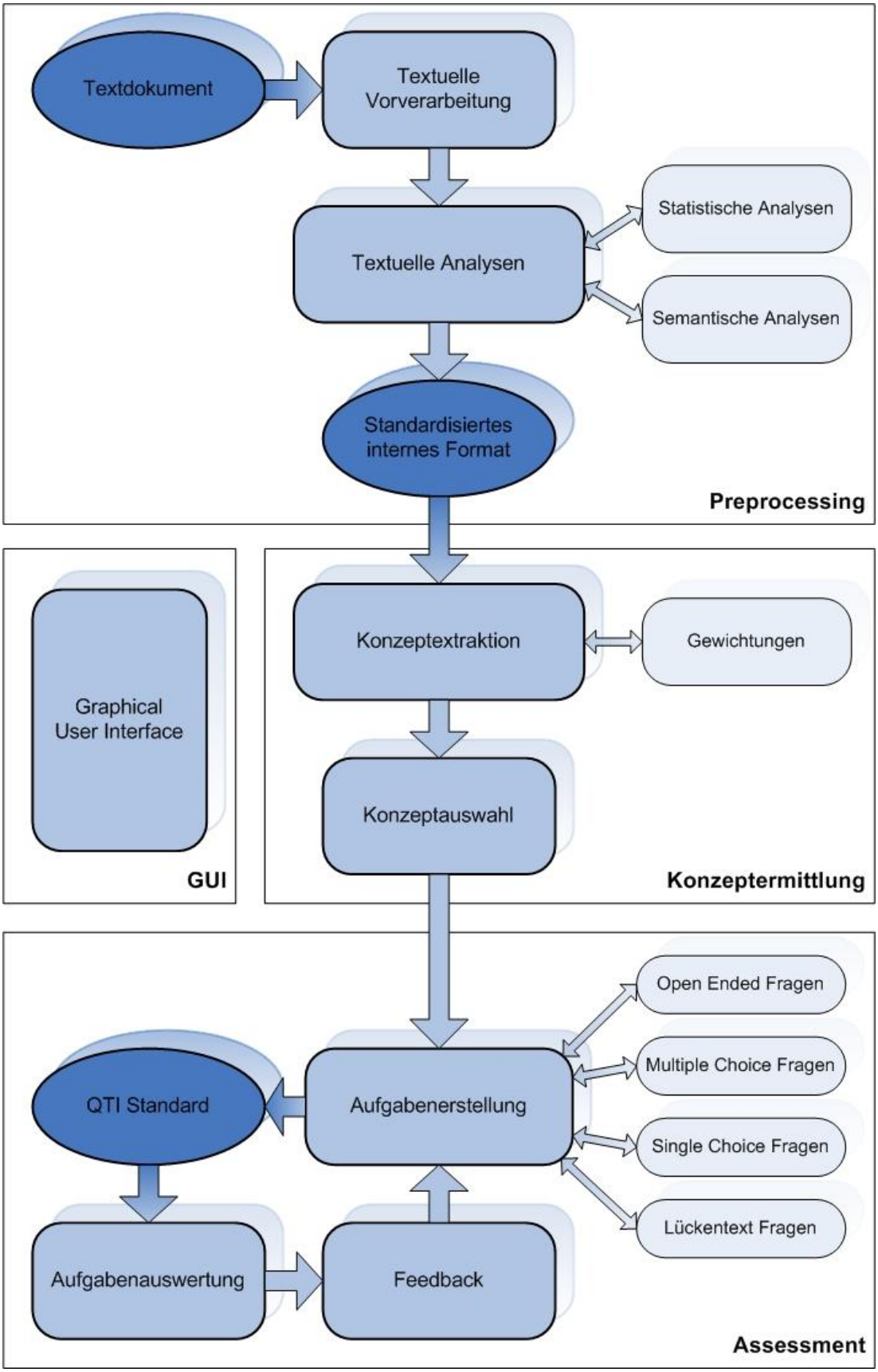


Abbildung 14: Konzeptionelles Design einer allgemeinen Lösung

4.2.1 Preprocessing

In diesem Subkapitel wird das grobe Design und die Funktionalität der einzelnen Komponenten des Preprocessing beschrieben und aufgezeigt, warum jene benötigt werden.

- **Textuelle Vorverarbeitung**

Die textuelle Vorverarbeitung umfasst das Überführen des Input Dokumentes, welches in einem der Standardformate wie Adobe PDF, Microsoft Word, HTML, XML, SGML, RTF, ODT, E-Mail und Plain Text vorliegen soll, in ein einheitliches Format welches in weiterer Folge die Grundlage für die nachfolgenden Module bildet.

- **Textuelle Analysen**

Die textuellen Analysen gliedern sich in statistische und semantische Analysen, bei welchen zu Beginn eine Tokenisierung, Stemming und ein Wortgewichtung durchgeführt werden. Anschließend werden ein Gazetteer und ein Sentence Splitter ausgeführt, um spezielle Annotationen und die Satzgrenzen zu detektierten. Ein zentrales Element ist das nachfolgende POS Tagging, bei welchem jedes Wort einer Wortklasse zugeordnet wird.

Nach dem POS Tagging wird ein Transducer ausgeführt und anhand dessen auf regulären Expressionen basierenden Muster auf die bisherigen Annotationen angewandt um in weiterer Folge eine Koreferenzauflösung zu gewährleisten. Abschließend ist es noch sinnvoll, einen Chunker auszuführen, da in weiterer Folge, nach der Bestimmung von einzelnen relevanten Wörtern, jene Phrasen in welchen diese inkludiert sind für das Erstellen von Fragen herangezogen werden können.

- **Standardisiertes internes Format**

Die Annotationen und Gewichte des vorhergehenden Schrittes werden erneut in ei einheitliches Format, optimalerweise in ein XML Format, welches neben dem Text jene im vorherigen Schritt gewonnen Informationen beinhaltet, übergeführt. Dies stellt einerseits sicher, dass die Inhalte jederzeit gespeichert und geladen werden können, andererseits aber auch, dass alle benötigten Informationen zentral verfügbar sind.

4.2.2 Konzeptermittlung

Die Konzeptermittlung besteht im Wesentlichen aus der Konzeptextraktion, wobei die wichtigsten Phrasen ermittelt werden, und aus der Konzeptauswahl, welche vom User beeinflusst werden können soll.

- **Konzeptextraktion**

Die Extraktion von wichtigen Wörtern, Phrasen und Konzepten kann, wie in Kapitel 3.1 anhand einiger interessanter Forschungsansätzen beschrieben wurde, auf viele Arten erfolgen. Im Wesentlichen gehen die meisten Ansätze auf Luhn zurück, der erstmals das Konzept der Worthäufigkeit eingeführt hat. Jene Häufigkeiten und Merkmale wie Wortähnlichkeiten werden dazu genutzt, um Zusammenhänge einzelner Phrase zu berechnen und daraus Schlüsse über die Wichtigkeit derer ziehen zu können.

- **Konzeptauswahl**

Die Konzeptauswahl besteht aus einer Interaktion des Users und den berechneten Schlüsselkonzepten, wobei der Nutzer die getroffene Auswahl beeinflussen kann.

4.2.3 Assessment

Das Assessment Modul besteht aus der eigentlichen Aufgabenerzeugung, dem Erzeugen von QTI Assessment Items, dem Auswerten von Fragen und dem Generieren von Feedback.

- **Aufgabenerstellung**

Die Aufgabenerstellung erzeugt aus den Schlüsselkonzepten beziehungsweise den Sätzen die jene beinhalten Open Ended, Single Choice, Multiple Choice und Lückentext Fragen. Wie genau das spezifische Design der einzelnen Fragetypen aussieht wird in Kapitel 6, in welchem die Implementierung beschrieben wird, abgehandelt und hier deshalb nicht näher erläutert.

- **Fragenauswertung**

Die Fragenauswertung, die je nach Fragetyp spezifische Vorgehensweisen erfordert, bietet die Möglichkeit eine vom Nutzer gegebene Antwort zu verifizieren und ist darüber hinaus die Vorraussetzung um automatisches Feedback erzeugen zu können. Nähere Informationen wie dies bewerkstelligt werden kann finden sich in Kapitel 6.6.4.

- **Feedback**

Das automatische Generieren von Feedback wird in Form von Einblendungen von relevanten Textpassagen, welche die erfragten Schlüsselkonzepte beinhalten, gegeben. Eine genauere Beschreibung einer möglichen Vorgehensweise findet sich in Kapitel 6.6.5.

4.2.4 GUI

Das Graphical User Interface bietet dem Nutzer die Möglichkeit sämtliche Einstellungen von Parametern vorzunehmen, den Auswahlprozess von Konzepten zu beeinflussen und die erzeugten Aufgaben darzustellen.

Die in diesem Kapitel dargelegten Konzepte verdeutlichen, im Sinne der Aufgabenstellung, einen vollständigen Ansatz. In der Implementierung werden Teile der textuellen Analysen, die Überführung in ein internes Dateiformat, Teile der Konzeptermittlung, die Konzeptauswahl und eine umfangreiche Aufgabenerstellung umgesetzt.

4.3 Zusammenfassung

In diesem Kapitel wurden die Anforderungen an ein System, das eine automatische Fragengenerierung umsetzt, aufgezeigt und herausgearbeitet, dass diese im Wesentlichen eine möglichst verallgemeinerbare, nicht themenspezifische Lösung, eine ausreichende textuelle Vorverarbeitung sowie eine geeignete Konzeptauswahl sind. Darüber hinaus sollen die Fragetypen Open Ended, Single Choice, Multiple Choice und Lückentexte unterstützt werden. Optimalerweise bietet ein derartiges System die Möglichkeit Antworten auf die generierten Fragen zu verifizieren und dem User Feedback zu geben. Eine Grundvoraussetzung für jene beiden Funktionen ist die Ermittlung einer Referenzantwort, welche mit der Fragengenerierung einhergeht.

Das Graphical User Interface soll intuitiv bedienbar sein und jeden Verarbeitungsschritt in geeigneter Art und Weise visualisieren. Weiters ist wichtig eine adäquate Benutzerinteraktion zu gewährleisten und dem User die Möglichkeit zu geben, anhand von Parametern den Gesamtprozess der Aufgabenerstellung zu beeinflussen. Zusätzlich soll auf gängige Standards geachtet und die erzeugten Informationen in XML Dateien und dem QTI Standard gespeichert werden.

Im folgenden Kapitel werden die in der Implementierung, dem Automatic Question Creator, verwendeten Tools und Frameworks vorgestellt. Dabei werden jene Tools und deren Funktionsweise allgemein betrachtet und vor Allem jene Komponenten erläutert, welche in der Implementierung eine konkrete Anwendung finden.

5 *Tools und Frameworks*

In diesem Kapitel werden die wichtigsten Tools beziehungsweise Frameworks vorgestellt, die für die Implementierung des Automatic Question Creators, welche in Kapitel 6 ausführlich beschrieben wird, eingesetzt werden. Da die Umsetzung des Automatic Question Creator in Java realisiert ist, um eine Unabhängigkeit von der verwendeten Plattform zu gewährleisten, sind die vorgestellten Tools beziehungsweise deren Frameworks ebenso in Java geschrieben. Es soll dabei weniger im Detail die gesamt mögliche Funktionalität der einzelnen Tools und Frameworks vorgestellt werden, sondern vielmehr jene Features, die in der praktischen Umsetzung dieser Arbeit Anwendung finden. Der Vollständigkeit halber wird dennoch kurz aufgezeigt, welche Features in den einzelnen Tools verfügbar sind.

5.1 GATE

Das GATE (general architecture for text engineering) Tool wurde entwickelt um die Weiterverarbeitung von Texten zu ermöglichen beziehungsweise zu optimieren. Dabei wird darauf geachtet, dass Aufgaben wie die Datenspeicherung, die Darstellung jener sowie die Komponenten zum Bearbeiten und Ausführen dieser strikt getrennt sind. Aufgrund dessen, durch die Implementierung in Java und die Einhaltung von XML Standards sind eine vollständige Austauschbarkeit und Erweiterbarkeit gewährleistet. Zusätzlich beinhaltet das Framework Maßnahmen zum Messen der Performance der einzelnen Komponenten (vgl. Cunningham, Maynard, Bontcheva und Tablan, 2002)

GATE akzeptiert laut Cunningham, Maynard, Bontcheva und Tablan (2002) als Input Dokumente der Formate XML, RTF; HTML, SGML, ODT, E-Mail und plain text. Im Benutzerhandbuch von Cunningham et al. (2009) sind darüber hinaus auch das Word und PDF Format aufgeführt, wobei angemerkt wird, dass nur Text und keine Formatierungseigenschaft ausgelesen werden kann. Eine Komponente in GATE kann einer der drei folgenden Arten angehören. Man spricht von einer *Language Resource*, also einem Lexikon, einem Korpus oder einer Ontologie, einer *Processing Resource*, beispielsweise einem Parser, oder einer *Visual Resource*, einer Komponente in der GUI. Ein Beispiel für das GATE GUI und die Art und Weise der Darstellung eines annotierten Korpus findet sich in Abbildung 15.

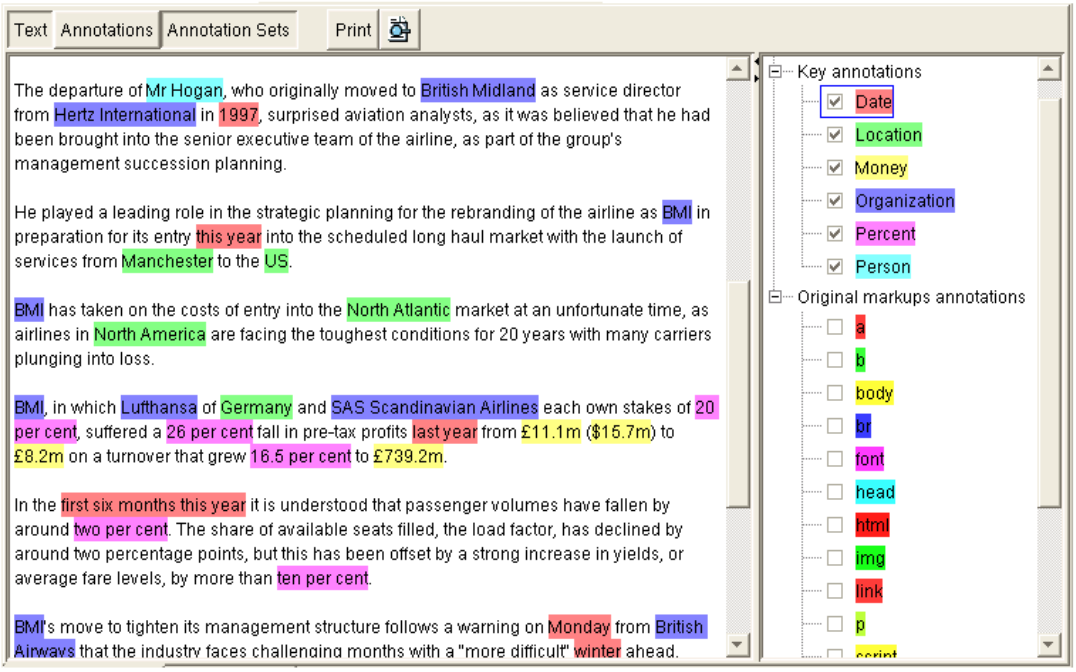


Abbildung 15: GATE GUI: Beispiel Informationsextraktion (vgl. GATE, 2010b)

5.1.1 Annotationsschema

Die Annotierung eines Textes und das Erzeugen eines Textkorpus ist die Grundlage für sämtliche weiterführende Operationen die das GATE Tool zur Verfügung stellt. Ein sehr einfaches, nicht vollständiges, aber demonstratives Beispiel ist in Tabelle 9 dargestellt. Es wird dabei aufgezeigt, dass zuerst die einzelnen Token eines Satzes mit deren ID, Start- und Endpunkten, sowie deren Features gespeichert werden. Anschließend werden zu jedem Token die besonderen Typen, in diesem Fall eine Person mit dem Verweis auf den Start und Endpunkt gehalten. Letztendlich beschreibt der Typ Sentence die Satzgrenzen mit dessen Start- und Endpunkten.

Text				
Cyndi savored the soup.				
^0...^5...^10..^15..^20				
Annotations				
Id	Type	SpanStart	Span End	Features
1	token	0	5	pos=NP
2	token	6	13	pos=VBD
3	token	14	17	pos=DT
4	token	18	22	pos=NN
5	token	22	23	
6	name	0	5	name_type=person
7	sentence	0	23	

Tabelle 9: Beispiel einer Gate Annotation (vgl. Cunningham et al., 2009)

5.1.2 Annotationstypen

Nachfolgend werden die einfachen und essentiellen Annotationstypen aufgezählt, welche mittels dem ANNIE Plug-in ohne Modifikationen erkannt werden können. Möglich sind das Erkennen von Personen, Orten, Organisationen, Nummern, Datumsangaben, Währungen, Sprachen, Kategorien, Jobtiteln, Adressen, Straßen, Koreferenzen, Telefonnummern, Einrichtungen, E-Mail und Internetadressen, Prozentangaben und Syntaxen.

Bewerkstelligt wird das Auffinden dieser Annotationstypen mittels Listen, die nach Bedarf einfach erweitert werden können. Neben Listen die, abgesehen von einer Vielzahl an vorgefertigten Einträgen, auch Regeln beinhalten um neue Einträge zu deklarieren, gibt es auch Listen mit Abkürzungen der genannten Einträge.

5.1.3 Plugins

Dieses Subkapitel stellt die wichtigsten, in GATE zur Verfügung stehenden, Plugins vor. Dabei werden keine genauen Vorgehensweisen, sondern vielmehr die Kernfunktionen beschrieben. Die Ausführung der Module erfolgt sequentiell in Form einer Pipeline.

- ANNIE: Das ANNIE, a Nearly-New Information Extraction System, Plug-in ist die GATE Hauptkomponente im Automatic Question Creator. Es bietet die Möglichkeit den Text zu Tokenisieren, einen Gazeteer auszuführen, die Satzgrenzen zu bestimmen, POS Tagging durchzuführen und Koreferenzen zu ermitteln. Die einzelnen Komponenten kommunizieren über das GATE Annotation API und nutzen finite-state Algorithmen und die JAPE Sprache (vgl. Cunningham et al., 2009). All die angeführten Funktionen finden bei der Implementierung Anwendung.

- CREOLE: Das CREOLE, a Collection of REusable Objects for Language Engineering, Plug-in, bietet eine Vielzahl von Processing Resources, Language Resources und Visual Resources. In der GATE Version 5 werden folgende CREOLE Plugins unterstützt: *Document Reset, Verb Group Chunker, Noun Phrase Chunker, Onto Text Gazetteer, Flexible Gazetteer, Gazetteer List Collector, Tree Tagger, Stemmer, GATE Morphological Analyzer, Mini-ParParser, RASP Parser, SUPPLE Parser, Stanford Parser, Montreal Parser, Language Plugins, Chemistry Tagger, Flexible Exporter, Annotation Set Transfer, Information Retrieval in GATE, Crawler, Google Plugin, Yahoo Plugin, WordNet in GATE, Machine Learning in GATE, MinorThird, MIAKT NLG Lexicon, Kea, Ontotext JapeC Compiler, ANNIC, Annotation Merging, OntoRoot Gazetteer* und *Chinese Word Segmentation* (vgl. Cunningham et al., 2009).

Mittels JAPE, a Java Annotation Patterns Engine, ist es möglich, auf die Annotationen beliebige reguläre Expressionen anzuwenden und auf diese Art und Weise eigene Annotationstypen zu implementieren.

5.2 WordNet

Das WordNet (2010) wurde 1985 an der Princeton Universität entwickelt und ist eine elektronische, lexikalische Datenbank für die englische Sprache. Nachfolgend wird kurz die Organisation der Einträge in WordNet erläutert und aufgezeigt, wie diese Einträge miteinander verknüpft sind. Nähere Informationen darüber, wie das WordNet bei der Implementierung im Detail eingesetzt wird, finden sich in den Kapiteln 6.2.4.2 und 6.2.4.3.

5.2.1 Aufbau

Es werden alle Wörter beziehungsweise Lemma als Knoten in den WordNet Baum eingeordnet und Informationen zu diesen gespeichert. Die sich daraus ergebende Hierarchie lässt sich in weiterer Folge dazu nutzen, Beziehungen abzubilden und daraus Wortähnlichkeiten zu berechnen. Jedes dieser Wörter wird in einem so genannten Synset gespeichert, worin die verschiedenen Ausprägungen von Wortarten in Synonymgruppen strukturiert werden und die jeweils als ein lexikalisches Konzept gesehen werden können. Es werden sowohl Wortbedeutungen und Definitionen als auch Relationen zwischen den jeweiligen Synsets gehalten. Durch diese Synsets ist es auch möglich, den durch die Baumstruktur fehlenden kontextuellen Zusammenhang auszudrücken (vgl. Tengi, 1998).

Aktuell beinhaltet das WordNet laut der offiziellen WordNet (2010) Homepage etwa 155.000 Wörter, 118.000 Synsets und 207.000 Wortpaare. Den größten Teil der Wörter nehmen, mit einer Anzahl von 118.000 Wörtern, die Hauptwörter, gefolgt von 11.500 Verben, 21.500 Adjektiven und 4.500 Adverbien ein.

Eine der größten Schwachstellen von WordNet ist die spärliche Integration von Eigennamen. Vor Allem im Fokus der Aufgabenstellung wäre es wichtig, bei der Bestimmung von ähnlichen Wörtern und der Distraktorenwahl auf eine Fülle von Eigennamen zurückgreifen zu können (vgl. Hearst, 1998).

5.2.2 Relationen

Die Wortbedeutungen beziehungsweise Synsets sind nach Hauptwörtern, Verben, Adjektiven und Adverbien gegliedert. Nachfolgend werden die einzelnen Abhängig-

keiten dieser Wortarten aufgeführt um in weiterer Folge einen Einblick in die Möglichkeiten von WordNet zu bekommen.

- Bei Hauptwörtern unterscheidet Miller (1998a) *hypernyms*, *hyponyms*, *coordinate terms*, *holonyms* und *meronyms*. Unter einem Hyperonym versteht man jeweils ein übergeordnetes, unter einem Hyponym ein untergeordnetes Synset. Als *coordinate term* werden dabei alle Hyponyme von einem Hyperonym bezeichnet. Ein Wort ist dann ein Holonym eines zweiten Wortes, wenn die Bedeutung dieses Wortes ein Teil der Bedeutung des vorliegenden Wortes ist. Ein Wort ist genau dann ein Meronym, wenn diese Bedingung umgedreht ist.

- Bei Verben werden laut Fellbaum (1998) *hypernymy, hyponymy*, *entailments*, *troponymy* und *coordinate terms* unterschieden. Ein Verb ist dann ein Hyponym eines anderen, wenn es eine Möglichkeit dessen ist, beziehungsweise ein Hyperonym, wenn es ein Überbegriff einer spezifischen Möglichkeit ist. Eine troponyme Relation liegt dann vor, wenn ein Verb eine spezifische Ausprägung eines anderen ist. Unter einem coordinate term versteht man wiederum alle hyponymen Verben der hypernymen Verben.

- Bei Adjektiven, die in deskriptive und relationale Adjektive unterteilt werden, gibt es laut Baek, Cho und Kim (2005) die konzeptionellen semantischen Beziehungen, die von dem Hauptwort, das in Verbindung mit dem Adjektiv auftritt, abhängen, sowie synonyme und antonyme Beziehungen.

- Die meisten Adverbien werden von Adjektiven abgeleitet, indem Suffixe angehängt werden. Es werden hierbei nur diese Ableitungen und synonyme sowie antonyme Relationen dargestellt. Jene Synonyme und Antonyme werden indirekt über die korrespondierenden Adjektive ausgewertet (vgl. Miller, 1998b).

Zusätzlich zu den obigen Möglichkeiten werden zu jedem Synset Beispiele der Verwendung gespeichert. Außerdem wird jedes Hauptwort einem von Miller (1998a) genannten *unique beginner* zugeordnet, worunter ein Synset verstanden wird, dass seinerseits kein hypernymes Synset hat und demnach eine Hauptwurzel darstellt. In Abbildung 16 werden die 25 *unique beginner* dargestellt, die nach den Files benannt sind, aus welchen die untergeordneten Hauptwörter entspringen. Manchmal werden anstatt der 25 vollständigen unique beginner nur elf von diesen herangezogen, wobei diese in der Abbildung kursiv hervorgehoben sind.

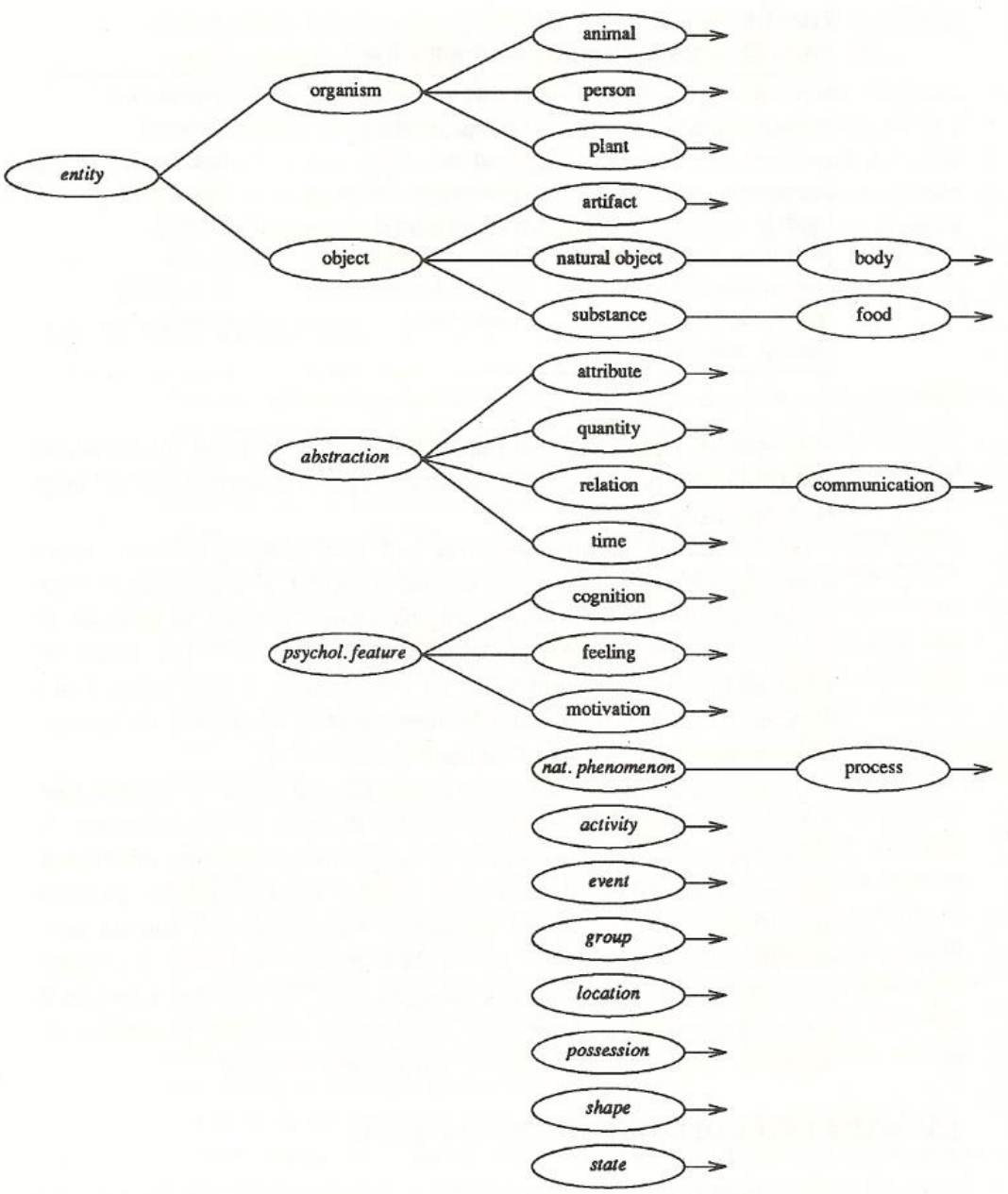


Abbildung 16: Unique beginner in WordNet (vgl. Miller, 1998a)

5.3 JQTI - QTI Standard

Die IMS Question and Test Interoperability Specification, die erstmals im Jahr 2000 veröffentlicht wurde und auf der QMS (Question Markup Language) aufbaut, beschreibt ein Standardformat für Assessment Inhalte und hat den großen Vorteil, auf verschiedenen Systemen einheitlich verwendbar zu sein. Der QTI Standard besteht aus einem Datenmodell und einem XML Schema, welche es in Kombination ermög-

lichen, einzelne Fragen, ganze Tests, aber auch die Ergebnisse auszutauschen und die Daten unabhängig von dem Tool, mit welchem diese erstellt wurden, zu sichern. Eingesetzt wird die QTI Spezifikation in *Learning Management Systems*, in *Test Content Management Systems*, Tools zur Aufgabenerstellung und in Kollektionen von Tests (vgl. Joosten-ten Brinke, Gorissen und Latour, 2005).

Beim grundsätzlichen Aufbau innerhalb dieser Spezifikation wird zwischen einem *Assessment Item*, das sowohl die eigentliche Frage, die Antwortmöglichkeiten als auch Informationen zur richtigen Antwort und zur Punktevergabe beinhaltet, und einem *Assessment*, welches aus einer Sammlung von Assessment Items besteht, unterschieden (vgl. Joosten-ten Brinke, Gorissen und Latour, 2005). Jedes dieser *Assessment Items* trägt einerseits Informationen über die Frage selbst, aber auch Informationen darüber, wie Antworten zu behandeln sind, in welcher Reihenfolge diese erscheinen und wie die Fragen formuliert sind (vgl. Miao, 2009).

Besonders erwähnenswert ist, dass ab QTI Spezifikation 2.0 der erhebliche Nachteil besteht jede Frage als eigenes File abzuspeichern zu müssen, was darin resultiert, dass die Erstellung eines tatsächliche Assessment Tests aus den so genannten Assessment Items ohne entsprechende Tools eine sehr zeitaufwändige Aufgabe darstellt.

5.4 Text Tiling

Der TextTiling Algorithmus von Hearst, der in Kapitel 2.1.6.5 bereits kurz beschrieben wurde, wird dazu genutzt, um basierend auf nicht strukturierten Textpassagen inhaltliche Sprünge zu markieren. Einerseits kann der Algorithmus zum Strukturieren eines Textes verwendet werden, andererseits aber auch um dazu, Passagen, in denen ein spezifisches Konzept auftritt, abzugrenzen. Im konkreten Fall kommt das TextTiling Modul des MorphAdorner Frameworks der Northwestern University (siehe MorphAdorner, 2008) zum Einsatz, um jene Textfragmente zu identifizieren, welche die jeweiligen Schlüsselkonzepte beinhalten.

5.5 XtraK4Me

Der Autor Schutz (2008) hat ein Java Framework entwickelt, welches es ermöglicht, anhand einiger statistischer Algorithmen, aufbauend auf einem annotierten Textkorpus, Schlüsselphrasen zu extrahieren. Der Algorithmus erfordert einen Text mit mindestens 500 Wörtern um brauchbare Ergebnisse zu liefern.

5.6 Synthetica

Als grafische Benutzeroberfläche kommt die Look and Feel Erweiterung Synthetica (siehe Synthetica, 2010) zum Einsatz, welche die grafische Darstellung der Java Oberfläche überschreibt. Dabei werden die BlueMoon, die BlackMoon und die SilverMoon, welche in nicht kommerziellen Anwendungen distributiert werden dürfen, in teilweise modifizierter Form verwendet.

5.7 Zusammenfassung

In diesem Kapitel wurden die wichtigsten Tools und Frameworks vorgestellt, die bei der begleitenden Implementierung zu dieser Arbeit eingesetzt werden. Es wird aufgezeigt, dass besonders GATE, welches für eine Annotierung verwendet werden kann, und WordNet, welches für die Bestimmung von Ähnlichkeiten und Distraktoren eingesetzt wird, zwei unverzichtbare Frameworks darstellen.

Zusätzlich wurde kurz der TextTiling Algorithmus, der Textfragmente zu zugehörigen Schlüsselphrasen identifizieren soll, vorgestellt und der QTI Standard sowie dessen wichtigste Bestandteile erläutert. Letztendlich wurde auch der ExtraK4Me Algorithmus, der in der Implementierung ebenfalls eingesetzt wird um relevante und vermeintlich wichtige Phrase zu ermitteln, beschrieben und kurz dargelegt, welche grafische Oberfläche für die Darstellung verwendet wird.

Im nächsten Kapitel wird der Automatic Question Creator vorgestellt und aufgezeigt, welche der bisher aus dieser Arbeit gewonnenen Erkenntnisse darin Anwendung finden und wie die in diesem Kapitel vorgestellten Hilfsmittel eingesetzt werden.

6 *Automatic Question Creator*

In diesem Kapitel wird die konkrete Implementierung vorgestellt, welche die wesentlichen Erkenntnisse aus dem theoretischen Teil umsetzt. Das Graphical User Interface und die interne Dokumentenstruktur wurden gemeinsam mit Weinhofer (2010) konzipiert, der in seiner Arbeit *Extraktion semantisch relevanter Daten aus natürlichsprachlichen Inhalten in Hinblick auf eine automatische Fragengenerierung* Konzepte entwickelt und implementiert hat, um aus beliebigen natürlichsprachlichen Dokumenten die wesentlichen Inhalte zu extrahieren. Der implementierte Ansatz nutzt neben den Phrasen die mit Hilfe des ExtraK4Me Algorithmus ermittelt wurden als Input die Ergebnisse der Arbeit des genannten Autors.

6.1 Konzeptionelles Design

Das grobe konzeptionelle Design der Aufgabenerstellung ist in Abbildung 17, welche die GATE Annotierung und das Überführen in ein internes Format zeigt, und Abbildung 18, welche den Gesamtprozess der Fragengenerierung illustriert, dargestellt und verdeutlicht den gesamten Programmablauf. Die einzelnen Programmkomponenten werden in Kapitel 6.2 genauer beschrieben.

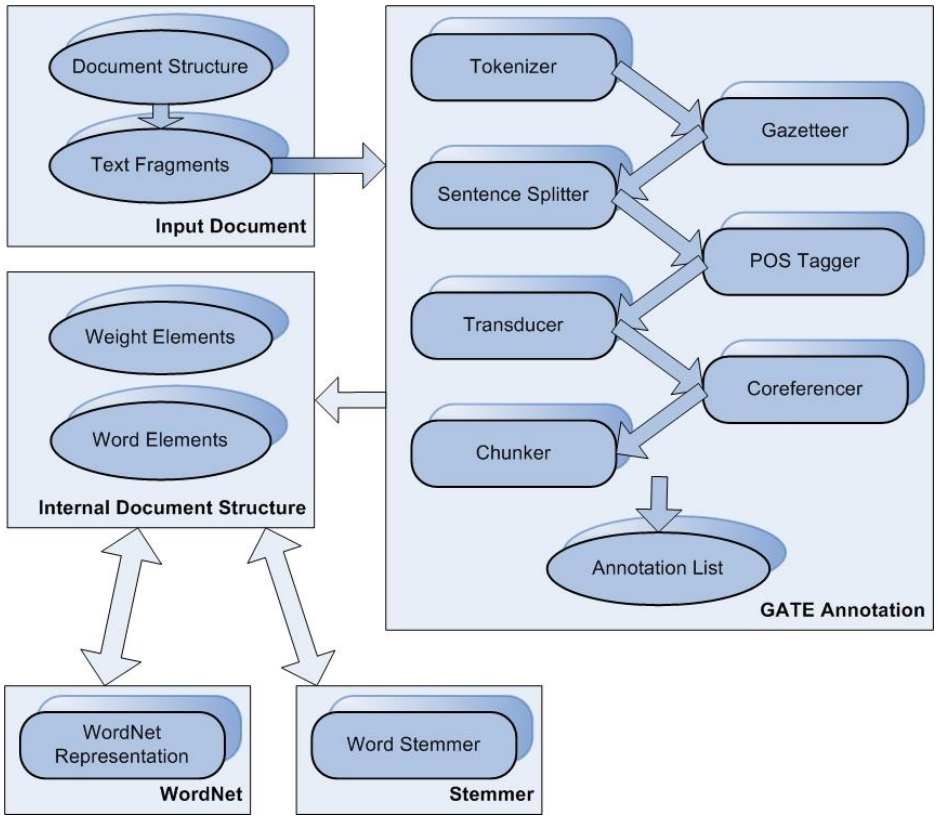


Abbildung 17: Preprocessing

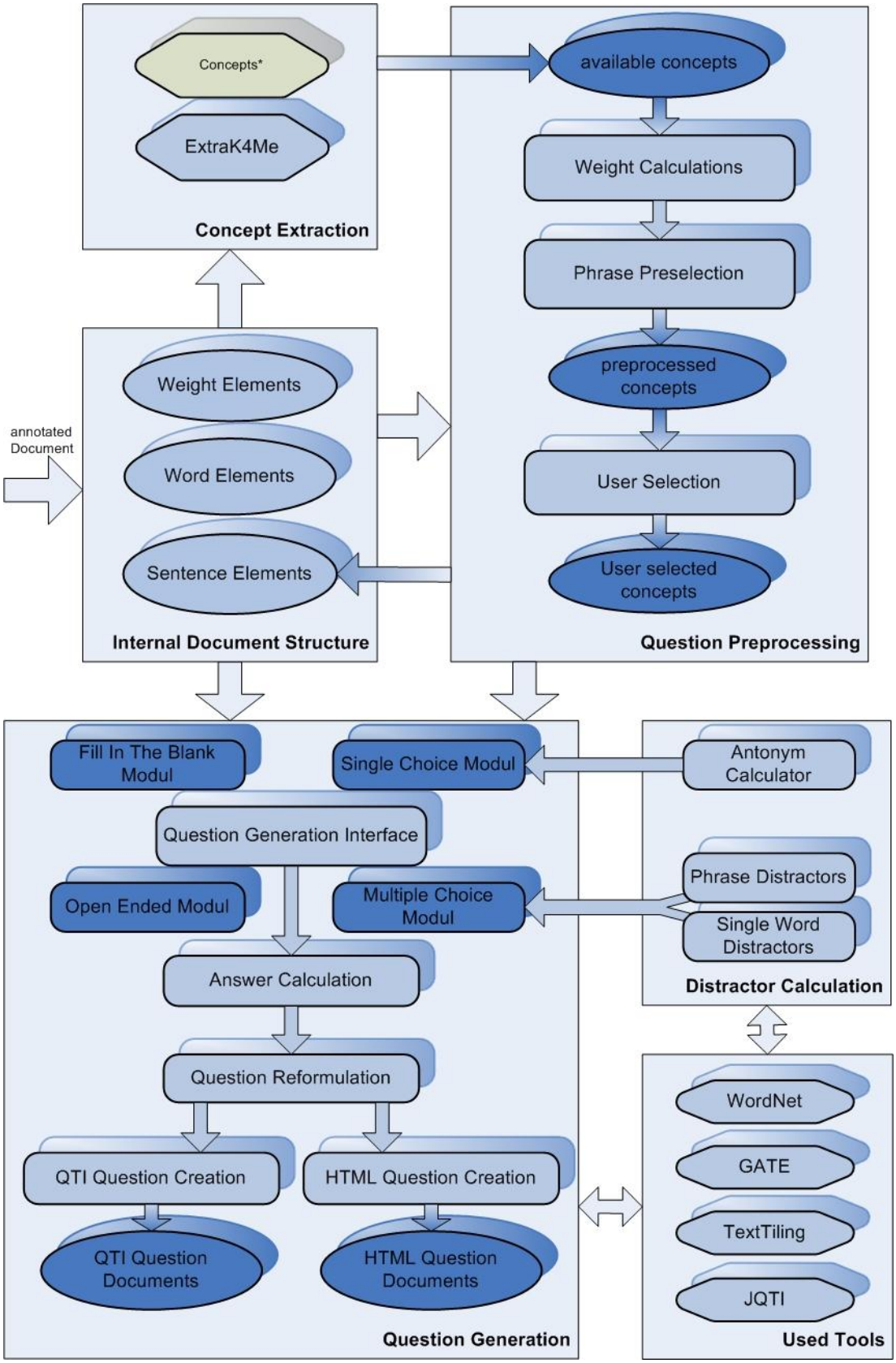


Abbildung 18: Konzeptionelles Design des Aufgabenerstellungsprozesses

6.2 Implementierung

In diesem Subkapitel werden die konzeptionellen Vorgehensweisen, welche in Abbildung 17 und Abbildung 18 abgebildet sind, detailliert erklärt. Zusätzlich wird erläutert in welcher Art und Weise jene Entwürfe zusammenhängen und wie die externen Tools eingebunden sind.

6.2.1 Preprocessing

Bevor die Konzepte zur Fragengenerierung implementiert werden können, ist es zu Beginn erforderlich, das Input Dokument zu annotieren und eine Datenstruktur einzuführen, die neben dem eigentlichen Text alle später benötigten Informationen speichern kann.

6.2.1.1 GATE Annotierung

Es wird eine GATE Annotierung mittels dem Annie Plugin in den Standardeinstellungen mit anschließendem Chunking durchgeführt. Die Annotation wird jedoch nicht auf das Gesamtdokument angewandt, sondern wird jeweils mit einzelnen Textfragmenten gestartet. Dies hat den Sinn und Zweck die ursprüngliche Formatierung beizubehalten, da es in weiterer Folge wichtig ist, zwischen normalen Sätzen, dem Titel, Überschriften und Ähnlichem unterscheiden zu können.

Die einzelnen Textfragmente werden durch die Annotation in eine so genannte Annotation List transformiert, welche aus Token und zusätzlichen speziellen Annotation besteht. Eine Beispielannotation wurde bereits in Kapitel 5.1 dargestellt, weshalb auf eine wiederholte Beschreibung des genauen Aufbaus verzichtet wird.

Aus jener Annotation List und den strukturellen Informationen aus dem Dokument werden die für sämtliche weitere Berechnungen benötigten eigenen Datenstrukturen erzeugt.

6.2.1.2 Interne Datenstruktur

Vorerst besteht die interne Datenrepräsentation aus zwei für dieses Projekt entwickelten Datenstruktur, den *Word Elements* und den *Weight Elements*.

- Ein so genanntes Word Element enthält eine eindeutige ID, das Wort selbst, dessen gestemmte Wortform, die WordNet Repräsentation des Wortes, den Unique Beginner in WordNet, Informationen über die Formatierung, die ID des Satzes in welchem das Wort auftritt, die ID der korrespondierenden Überschrift, eine Liste ähnlicher Wörter und deren IDs, eine Liste mit dem

Ähnlichkeitsmaß, eine Liste mit den Gewichten der ähnlichen Wörter, die IDs aller Wörter im korrespondierenden Chunk, den POS Tag und letztendlich spezielle Annotationstypen.

- Ein so genanntes Weight Element, enthält neben der ID, das statistische Gewicht, Faktoren einer Wortähnlichkeitsberechnung, der Ähnlichkeit zum Abstract, den Keywords, dem Titel und zu Überschriften, ein Gewicht für spezielle Annotationstypen, Kategorien und Formatierungen, das Phrasengewicht und letztendlich ein Gewicht für rekursive Ähnlichkeitsberechnungen.

Es ist außerdem wichtig zu erwähnen, dass es für jedes Word Element ein zugehöriges Weight Element gibt. Die Gewichtsberechnungen werden von den Programmteilen des Automatic Question Creators durchgeführt, welche begleitend zu einer parallel zu dieser Arbeit geschriebenen Masterarbeit von Weinhofer (2010) implementiert wurden und dessen Ergebnisse genutzt werden können. Es wurden von Beginn an diese gemeinsamen Datenstrukturen genutzt, um einerseits zu gewährleisten, dass sämtliche Programmmodule die selben Daten erhalten und manipulieren und andererseits, damit alle relevanten Informationen zentral gespeichert sind, auch wenn in einigen Fällen nicht alle Daten benötigt werden.

6.2.2 Concept Extraction

Zu Beginn der Konzeptextraktion wird der Text jedes Kapitels, wobei nur der Text und Aufzählungspunkte, nicht aber Überschriften oder Ähnliches beachtet werden, extrahiert. Anschließend wird der XtraK4Me Algorithmus mit jedem dieser Textfragmente aufgerufen und liefert in Form einer Liste die vermeintlich relevanten Schlüsselphrasen zurück. Abschließend werden in den Weight Elements der entsprechenden Wörter die gemäßen Eintragungen vorgenommen.

Jene Erkenntnisse und die Ergebnisse der bereits erwähnten Arbeit von Weinhofer (2010) werden kombiniert und in Form einer nach Gewicht der einzelnen Schlüsselwörter gereihten Liste zwischengespeichert.

6.2.3 Question Preprocessing

Die eingehenden Daten im Question Preprocessing Modul sind alle verfügbaren Konzepte in Form einer Liste mit den einzelnen Wörtern. Jene ist nach dem ermittelten und normierten Gewicht der Wörter und Phrasen geordnet. Diese Ordnung erfolgt kapitelweise, damit in jedem Kapitel die relevantesten Konzepte ausgewählt werden können und in kürzeren Kapiteln keine Starvation von Phrasen auftritt, welche sich aus geringeren Gewichten gegenüber längeren Kapiteln ergeben würde.

6.2.3.1 Weight Calculation

Zu Beginn werden die einzelnen Wortgewichte sämtlicher Sätze jedes Kapitels aufsummiert, es werden also Satzgewichte eingeführt. Beginnend mit dem am höchsten gewichteten Wort wird für jedes dieser Wörter der in Summe am höchsten gewichtete Satz berechnet, in welchem das Wort auftritt. Zusätzlich werden jene Phrasen extrahiert in welchen die Wörter in den jeweils am höchsten gewichteten Sätzen enthalten sind. Jeder der ausgewählten Sätze wird in einer neuen, zusätzlichen Datenstruktur, einem Sentence Element gespeichert.

Ein Sentence Element enthält die jeweilige Satz ID, die ID des ersten Wortes, jene des letzten Wortes, das Satzgewicht und eine Liste aller Wörter in der Phrase, welche in diesem Satz erfragt wird sollen.

6.2.3.2 Phrase Preselection

Um die Auswahl an Fragephrasen und Wörtern etwas einzugrenzen, werden für jedes Kapitel der Median der Phrasengewichte errechnet und nur die bessere Hälfte der Phrasen und die zugehörigen Sätze beibehalten. Im nächsten Schritt werden dem User für jedes Kapitel die nach Gewicht gereihten Phrasen und die Wörter, aufgrund welcher jene Phrase ausgewählt wurde, angezeigt. Darüber hinaus hat der User die Möglichkeit, einzelne Phrasen, welche sich nicht in der getroffenen Vorauswahl befinden, hinzuzufügen oder ausgewählte Phrasen auszuklammern und somit den Prozess der Konzeptauswahl aktiv zu beeinflussen.

6.2.4 Question Generation

Das Question Generation Modul implementiert den eigentlichen Prozess des Fragengenerierens. Es ist die Schnittstelle für das Open Ended, das Fill In The Blank, das Single Choice und das Multiple Choice Modul und wird darüber hinaus genutzt unerwünschte Zeichenfolgen zu eliminieren. Zusätzlich werden darin die für die eigentlichen Fragen benötigten Verzeichnisse im Dateisystem erstellt, bei Bedarf Pluralformen für Wörter im Singular gebildet, die Namen der Kapitel ausgelesen und Satzumgebungen berechnet. Die Berechnungen der Satzumgebung dienen als Hilfestellung bei Fill In The Blank und Multiple Choice Fragen, um einen gewissen Kontext in die Frage zu bringen, indem jeweils ein Satz vor und nach dem erfragten Satz eingeblendet wird. Bei Bedarf, wenn der Fragesatz also der Erste oder Letzte in einem Absatz ist, werden zwei Sätze davor oder danach extrahiert.

6.2.4.1 Fill In The Blank Modul

Das Fill In The Blank Modul ist vom Aufbau her recht einfach gehalten, da es das Wesen von Fill In The Blank Fragen nur erfordert Schlüsselkonzepte in einer Text-

passage auszublenden. Es wird demnach die nach Gewicht der Fragephrasen gereihte Liste abgearbeitet und in jedem der zugehörigen Sätze, inklusive jener Sätze, welche die bereits ermittelte Satzumgebung darstellen, alle Vorkommen der Phrase durch eine Lücke ersetzt. Es ist hierbei besonders auf die Groß- und Kleinschreibung und die Eliminierung von unerwünschten Zeichen in der Phrase und dem Satz zu achten, da ansonsten bei der Erstellung der Lücken Probleme auftreten können.

Die modifizierten Sätze und die Antworten, also die korrekten Phrasen, welche in die Lücken eingesetzt werden müssen, werden abschließend zur Weiterverarbeitung im Question Generation Modul zwischengespeichert.

6.2.4.2 Multiple Choice Modul

Das Multiple Choice Modul wurde vom Prinzip her ähnlich gestaltet wie das Fill In The Blank Modul, wobei die große Herausforderung bei diesem Fragetyp das Auffinden von geeigneten Distraktoren, also den falschen vorgegebenen Antworten ist. Die zu erfragende Phrase muss zumindest POS getaggt sein, da die Antwortmöglichkeiten vom Worttyp abhängig sind, sofern die Distraktoren mit Hilfe von WordNet bestimmt werden. Für jede Frage werden vorerst alle möglichen Distraktoren berechnet und auf eine Liste gespeichert, wobei sich Vorgehensweise für das Auffinden von Distraktoren in mehrere Stufen gliedert.

Grundsätzlich werden alle speziellen Annotation wie Orte, Personen, etc. bevorzugt und gesondert behandelt, da WordNet dahingehend Schwächen aufweist, weil derartige Wörter oftmals nicht in der WordNet Hierarchie enthalten sind. Beinhaltet eine Fragephrase demnach eine oder mehrerer spezielle Annotationen so werden alle Alternativen, also alle speziellen Annotationen des selben Types, im vorliegenden Dokument berechnet und in einer Liste gespeichert. Anschließend wird bei mehreren Annotationstypen, sofern für jeden zumindest drei Distraktoren verfügbar sind, per Zufall ein Typ ausgewählt und jene korrespondieren Alternativen auf die Distraktorenliste geschrieben.

Sofern keine spezielle Annotation oder zu wenige Distraktoren für die enthaltenen Annotationstypen zur Verfügung stehen wird die Distraktorenkalkulation mit WordNet bewerkstelligt. In diesem Fall wird eingangs versucht das Synset für die gesamte Fragephrase zu erhalten. Sollte die ganze Phrase in WordNet vorhanden sein, so können die so genannten coordinate terms, also die Hyponyme von den Hypernymen als potentielle Distraktoren in die zugehörige Distraktorenliste geschrieben werden.

Etwas komplizierter ist der am häufigsten auftretende Fall, dass die gesamte Phrase nicht eins zu eins in WordNet enthalten ist. Hierbei wird als nächstes versucht, die längste Teilphrase zu bilden, welche eine WordNet Repräsentation auf-

weist. Dies geschieht durch Abarbeitung sämtlicher Möglichkeiten von Teilphrasen, wobei beispielsweise bei einer Phrasenlänge von vier zuerst alle Phrase der Länge drei, dann alle der Länge zwei gebildet werden. In jedem Iterationsschritt wird versucht, zu jenen Phrasen ein Index Word in WordNet zu erhalten. Es werden also jene Distraktoren bevorzugt und auf die Distraktorenliste geschrieben, welche coordinate terms zur längsten Phrase darstellen.

Wenn noch immer kein Synset in WordNet ermittelt werden konnte, werden für alle einzelnen Wörter der Phrase, welche eine WordNet Repräsentation beinhalten, alle möglichen Distraktoren ermittelt und per Zufall eine der kalkulierten Listen mit mehr als drei Distraktoren ausgewählt.

Zu diesem Zeitpunkt werden jene Fragen ausgeklammert, wessen Fragephrase keine der obigen Bedingungen erfüllt hat, für welche also nicht genügend Distraktoren ermittelt werden konnten. Im nächsten Schritt werden im Falle von speziellen Annotationen die Lücken neu errechnet, indem die nicht zur speziellen Annotation gehörigen Phrasenteile wieder in den Satz eingegliedert werden. Dasselbe Prinzip wird bei mit Hilfe von WordNet berechneten Distraktoren durchgeführt, wobei immer jener Teil der Fragephrase, welcher nicht Bestandteil der Ausgangsphrase für die WordNet Berechnungen ist, wieder im Satz eingesetzt werden muss. Die Distraktorenauswahl unterliegt zusätzlich der Einschränkung, dass für mehrteilige Phrasen auch mehrteilige Distraktoren bevorzugt werden, um authentischere Antwortmöglichkeiten vorzugeben.

Um die Qualität der Distraktoren zu steigern werden in allen Fällen die Groß- und Kleinschreibung in Abhängigkeit von der richtigen Lösung für die Distraktoren übernommen. Darüber hinaus wird mittels sehr einfachen Regeln geprüft, ob das letzte Wort der tatsächlichen Fragephrase im Plural auftritt, da in diesem Fall das letzte Wort der Distraktoren ebenso auf die Pluralform zu bringen ist.

Unabhängig von der Ermittlungsart werden immer drei Distraktoren per Zufall ausgewählt und gemeinsam mit der Fragenliste und der Antwortliste im Question Generation Modul gespeichert.

6.2.4.3 Single Choice Modul

Bei Single Choice Fragen ist es das Ziel, Schlüsselsätze durch Austauschen beziehungsweise Weglassen eines Wortes oder einer Phrase so zu manipulieren, dass der Sinn des ursprünglichen Satzes verändert wird. Im Unterschied zu der Vorgehensweise bei der Ermittlung von Multiple Choice Fragen werden in diesem Modul die Kalkulationen von WordNet im Vergleich zu jenen von speziellen Annotationen bevorzugt.

Analog dem Prozess der Distraktorensuche in WordNet bei Multiple Choice Fragen werden bei Single Choice Fragen mit Hilfe von WordNet Antonyme berechnet und alle möglichen Antonyme für alle Wörter der Fragephrase berechnet. Für diese Berechnung werden Hauptwörter und Adjektive herangezogen, auf Antonyme von Verben wurde aufgrund von sehr unbefriedigenden Ergebnissen verzichtet. Gelingen das Auffinden von Antonymen, werden diese vorerst zwischengespeichert.

Sollte obiger Versuch keine Distraktoren liefern wird als nächstes geprüft, ob die Phrase spezielle Annotationen beinhaltet. Ist dies zutreffend, so können sämtliche spezielle Annotationen gleichen Types als Distraktoren fungieren. Im Falle eines positiven Ergebnisses werden jene in eine Liste von potentiellen Austauschkandidaten geschrieben.

Sofern auch jene Methode keine Möglichkeiten bietet einem Satz einen verfälschten Ausdruck zu verleihen, werden für alle Wörter des ganzen Satzes, in welchem die Phrase enthalten ist, Antonyme in WordNet berechnet. Diese Maßnahme birgt aufgrund der Änderung des erfragten Konzeptes zwar die Gefahr, Relevanz zu verlieren, es wurde jedoch festgestellt, dass die zwei zu Beginn geschilderten Berechnungsarten zu selten Antonyme beziehungsweise austauschbare Phrasen liefern.

Abschließend wird in jenen Sätzen ein Wort beziehungsweise ein Phrase, für welche Antonyme gefunden wurden mit einem von jenen per Zufall ausgewählten ersetzt und der User in weiterer Folge dadurch vor eine ja/nein Entscheidung gestellt. Wie im Falle aller Fragetypen werden erneut die Frage- und die Antwortliste im Question Generation Modul eingetragen.

6.2.4.4 Open Ended Modul

Das Open Ended Modul berücksichtigt bei der Erstellung von Aufgaben vordergründig die Art der Phrase beziehungsweise die darin beinhalteten speziellen Annotationen. Es wird darauf aufbauend zu Beginn unterschieden, ob eine Fragephrase ein von GATE erkanntes Wort eines speziellen Types, im konkreten eine Person, eine Organisation, einen Ort oder eine Datumsangabe, beinhaltet. Der Vollständigkeit halber sei erwähnt, dass eine spezielle Annotation auch aus mehreren Wörtern bestehen kann.

Wird in der Phrase genau eine spezielle Annotation beinhaltet, so werden die Fragesätze nach für den Annotationstyp spezifischen Mustern gebildet. Nachfolgend werden für jeden der implementierten Annotationstypen die einfach gehaltenen Muster vorgestellt. Hierbei ist zu beachten, dass nicht die komplette Phrase erfragt wird, sondern nur der jeweilige Teil einer speziellen Annotation.

- Person
  “What do you know about <Person> and what did that Person do in the focus of <ChapterName>?”
- Organisation
  “What is/was the role of <Organization> related to <ChapterName>?”
- Ort
  “What do you know about <Location> related to <ChapterName>?”
- Zeitangabe
  “What happened in <Date> and what do you know about that happening in the context of <ChapterName>?”

Auf die Auswertung von mehreren speziellen Annotationstypen in einer Phrase wurde vorerst verzichtet, Erweiterungsvorschläge dahingehend finden sich in Kapitel 6.6.1. Enthält eine Fragephrase demnach keine oder mehrere spezielle Annotationen wird ein Standardmuster der Form „What do you know about <Phrase> in the context of <ChapterName>?“.

Die erstellten Fragen werden im Question Generation Modul zwischengespeichert, auf die Art und Weise der Ermittlung der Referenzantwort wird nachfolgend eingegangen.

Die Kalkulation der Referenzantwort, welche nur bei Open Ended Fragen gesondert behandelt werden muss, wird mit Hilfe des bereits beschriebenen TextTiling Algorithmus vom MorphAdorner Framework durchgeführt. Für den gesamten Text jedes Kapitels werden die so genannten Textsegmentgrenzen berechnet und die jeweiligen Textfragmente, welche immer eine gewisse Anzahl an vollständigen Sätzen darstellen, gespeichert. Anschließend wird jener Textblock gesucht, welcher den zu der Fragephrase gehörigen Satz enthält und zu jener Frage in der korrespondierenden Antwortliste eingetragen. Jene Referenzantwort könnte in weiterer Folge zur Auswertung der gegebenen Antwort, genutzt werden (siehe auch Kapitel 6.6.4).

6.2.5 HTML Question Creation

Das HTML Question Creation Modul erzeugt für jeden Fragetyp ein eigenes HTML File, welches sämtliche Fragen dieses Types enthält. Als Input dienen je nach Fragetyp die Listen mit den Fragen, den Antworten und wenn nötig den Distraktoren, welche allesamt nach den einzelnen Kapiteln gegliedert sind. Wie jene HTML Files im Detail aussehen ist in Kapitel 6.4 ersichtlich.

6.2.6 QTI Question Creation

Analog dem HTML Question Creation Modul erstellt das QTI Question Creation Modul basierend auf den Listen welche die Fragen, die Antworten und die Distraktoren beinhalten, sämtliche Fragen für jeden Fragetyp. Es wird anhand des JQTI (siehe auch QTI, 2008) Frameworks für jede Frage ein Assessment Item erstellt und in ein eigenes XML File gespeichert. Der genaue Aufbau eines Assessment Items ist natürlich vom Fragetyp abhängig und wird hierbei nicht näher erläutert, da relativ viele XML Tags zu erzeugen sind. Obwohl im Zuge der Implementierung auf eine Fragenauswertung beziehungsweise Bewertung verzichtet wurde, sind alle nötigen QTI Informationen implementiert um eine vollständige Weiterverarbeitung der QTI Items in anderen Systemen zu ermöglichen. Die Darstellung von Fragen im QTI Standard findet sich ebenso in Kapitel 6.4.

6.3 Probleme der Umsetzung

In der Umsetzung der Implementierung traten vor allem Probleme mit WordNet, welche auf die bereits genannte Schwäche von unzureichender Unterstützung von Eigennamen zurückzuführen sind. Für jene Wörter, welche über keine Repräsentation in WordNet Verfügen, wurde aufgrund dessen bei der Distraktorenberechnung auf Wörter gleichen Types innerhalb des Dokumentes zurückgegriffen.

Eine weitere Schwachstelle kann GATE darstellen, da einige Male falsche beziehungsweise unvollständige Annotationstypen zugewiesen werden. Obwohl dieses Problem vorwiegend die Konzeptextraktion betrifft, wirkt sich dieser Fehler unweigerlich auf die Aufgabenerstellung aus, da jene Annotationen in die Muster zur Fragenerzeugung eingesetzt werden.

Bezüglich des Laufzeitverhaltens ist anzumerken, dass bei einem fünfseitigen Dokument, die Aufgabenerstellung circa 30 Sekunden in Anspruch nimmt. Diese Laufzeit resultiert aus dem Umstand, dass, je nach vorgenommenen Parametereinstellungen, auf dem Dateisystem für jede einzelne Frage eine HTML und eine XML Datei angelegt werden, wodurch im Durchschnitt 400 Dateien erzeugt werden müssen. Die eigentliche Berechnung der Fragen geschieht mehr oder weniger in Echtzeit, die schilderten Laufzeitverhalten sind durch die Anforderungen des QTI Standards, wonach jedes Assessment Item in ein XML File geschrieben werden muss, bedingt.

Allgemein anzumerken ist der Umstand, dass die Dokumentation der Frameworks einiger verwendeten Tools unvollständig ist und Schwierigkeiten bei der Integration in das vorliegende System auftraten.

6.4 Sichtweise des Nutzers

In diesem Subkapitel wird die Sichtweise des Nutzers bei der Nutzung des Automatic Question Creators, welcher in seiner Gesamtfunktionalität wie bereits erwähnt in Kooperation mit Weinhofer (2010) erstellt wurde, dargelegt.

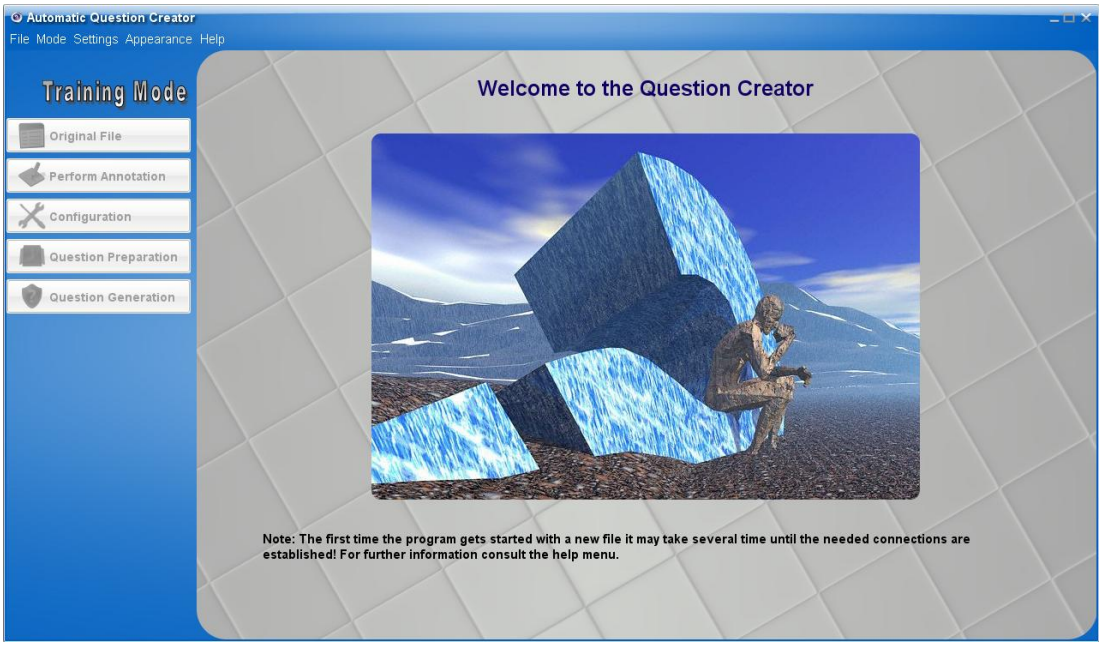


Abbildung 19: Oberfläche des Automatic Question Creators

Der Startbildschirm des Automatic Question Creators ist in Abbildung 19 dargestellt, wobei ersichtlich ist, dass die Buttons der Navigationsleiste grau unterlegt sind, da jene erst jeweils nach den nötigen sequentiellen Verarbeitungsschritten freigeschaltet werden. Das File Menü bietet die Möglichkeit ein neues File einzulesen, eine URL (vorzugsweise Wikipedia) anzugeben, ein annotiertes File, Settings und gewichtete Dokumente zu laden und zu speichern. Im Settings Menü, welches in Abbildung 20 ersichtlich ist, kann der User sämtliche Parameter einstellen, welche den Prozess der Konzeptextraktion und der Aufgabenerstellung beeinflussen.

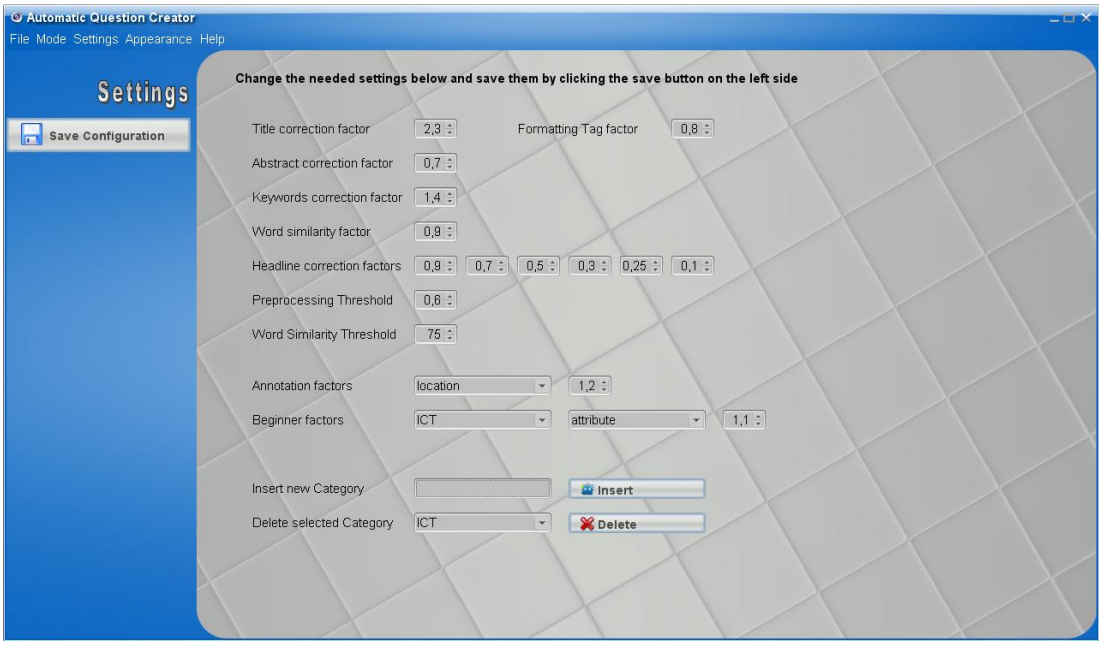


Abbildung 20: Settings Menü

Abbildung 21: GATE Annotation und Gewichtung

In Abbildung 21 ist ein annotierter und bereits gewichteter Beispieltext dargestellt. Die speziellen Annotationen, welche anhand von GATE berechnet wurden, sind farblich hervorgehoben und in einer Legende erklärt. Wenn der Mauszeiger über ein Wort des gezeigten Textes bewegt wird öffnet sich ein kleines Popupfenster, welches Informationen über die bereits durchgeführte Gewichtung liefert. Mittels dieser Information kann der User nachvollziehen, warum die im nachfolgenden

Schritt extrahierten Konzepte ausgewählt wurden. Eine genaue Erklärung der Funktionalität der Gewichtung liefert Weinhofer (2010) in seiner Masterarbeit.

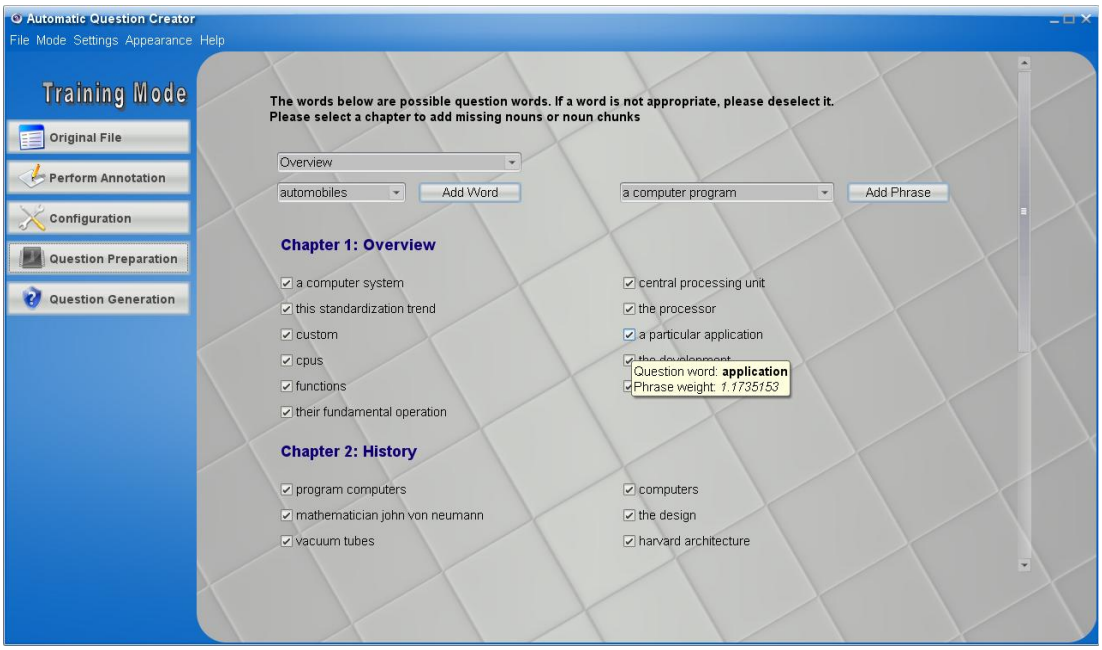


Abbildung 22: Question Preprocessing

In Abbildung 22 ist die Vorauswahl von verfügbaren Konzepten im Beispieltext aufgezeigt. Darüber hinaus ist ersichtlich, dass der User bei Bedarf erneut per Popupfenster angezeigt bekommt, aufgrund welchen Wortes eine spezifische Schlüsselphrase ausgewählt wurde. Zusätzlich wird aufgezeigt, dass über drei DropDown Menüs nicht berücksichtigte Wörter und Phrasen nachträglich ausgewählt werden können. Das erste jener Menüs erlaubt die Auswahl eines Kapitels, das zweite die Auswahl eines einzelnen Hauptwortes und das dritte Menü die Auswahl einer kompletten Phrase im gewählten Kapitel des Textes.

Nachdem der User mit der getroffenen Vorauswahl zufrieden ist beziehungsweise die erwünschten Terme hinzugefügt hat kann die Aufgabenerstellung durchgeführt werden. Es werden dabei immer alle verfügbaren Fragen sämtlicher Fragetypen berechnet und die daraus erzeugten Files in den entsprechenden Ordnern im Dateisystem abgelegt. In Abbildung 23 ist jener Bildschirm abgebildet, welcher nach Berechnung der Fragen angezeigt wird. Der User kann im nächsten Schritt die generierten Aufgaben der spezifischen Fragetypen mittels dem eigenen HTML File, welches alle Fragen eines Types gemeinsam anzeigt, oder die erzeugten Fragen im QTI Standard einzeln ansehen. Die Umschaltung der Ansicht erfolgt über die Buttons in der Navigationsleiste.

Abbildung 23: Question Generation Oberfläche

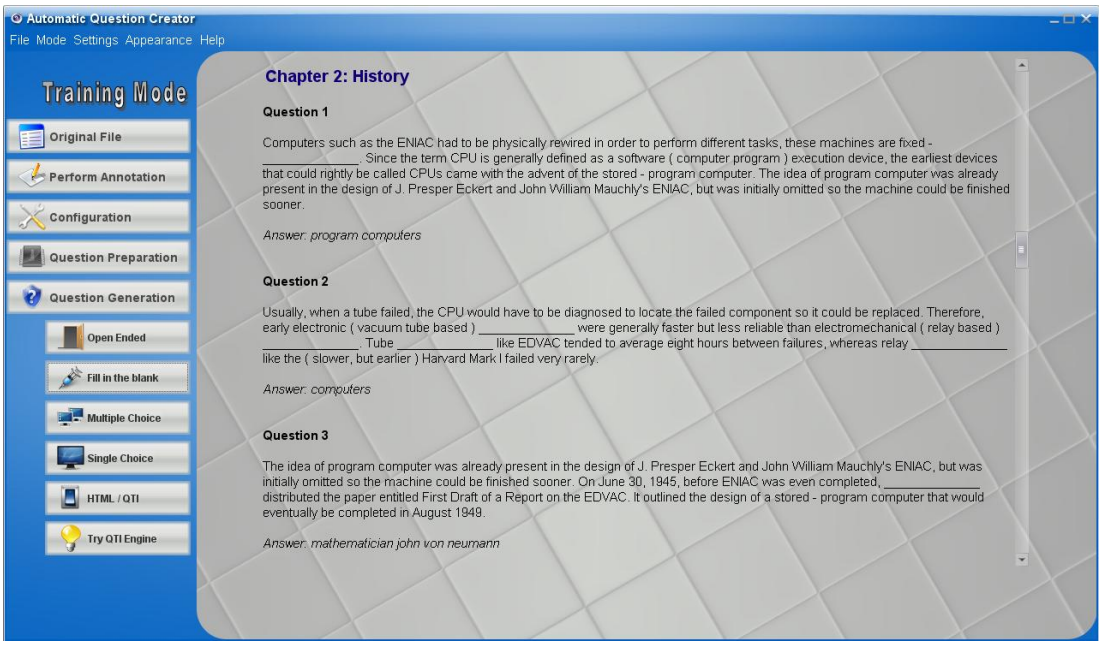


Abbildung 24: Fill In The Blank HTML Darstellung

Abbildung 24 zeigt die Ansicht der erzeugten Fill In The Blank Fragen im HTML Format, in welchem alle Fragen dieses Typs dargestellt werden.

In Abbildung 25 ist die Visualisierung einer Open Ended Frage im QTI Standard aufgezeigt. Neben der eigentlichen Frage werden ein Textfeld und die kalkulierte Referenzantwort angezeigt. Im Textfeld könnte in weiterer Folge vom User eine

Antwort eingegeben werden, die Auswertung wurde wie bereits erwähnt jedoch nicht implementiert. Die einzelnen QTI Files aller Fragetypen können mittels der QTI Engine (Link: http://qtiengine.qtitools.org/) getestet werden, die Auswertung bei Open Ended Fragen ist in der QTI Engine natürlich nicht implementiert.

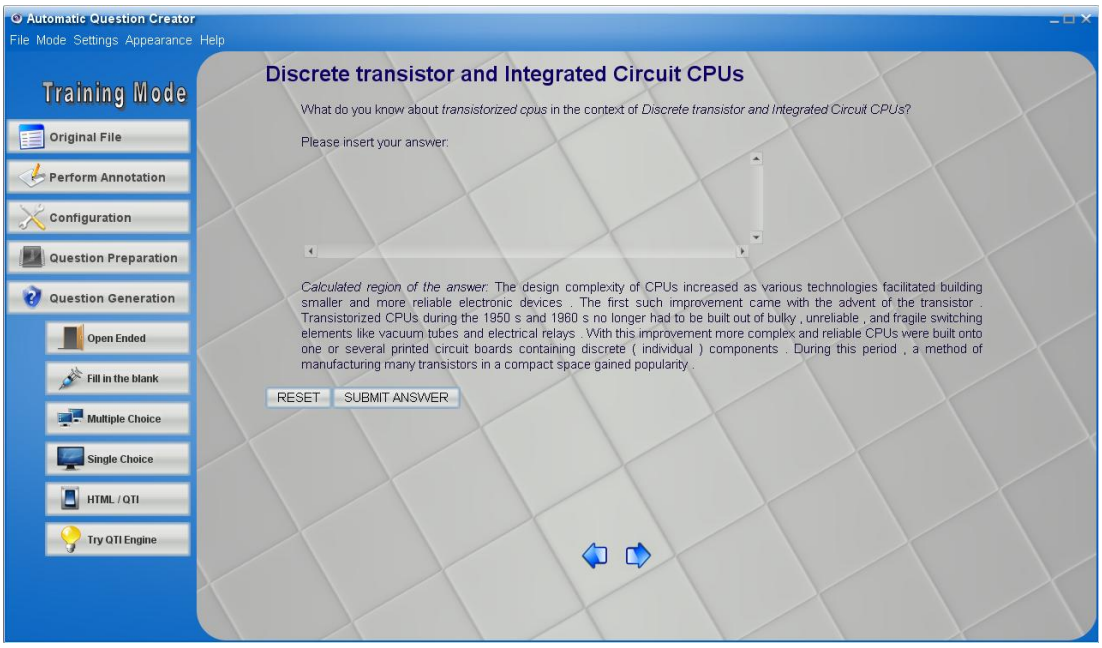


Abbildung 25: Open Ended QTI Darstellung

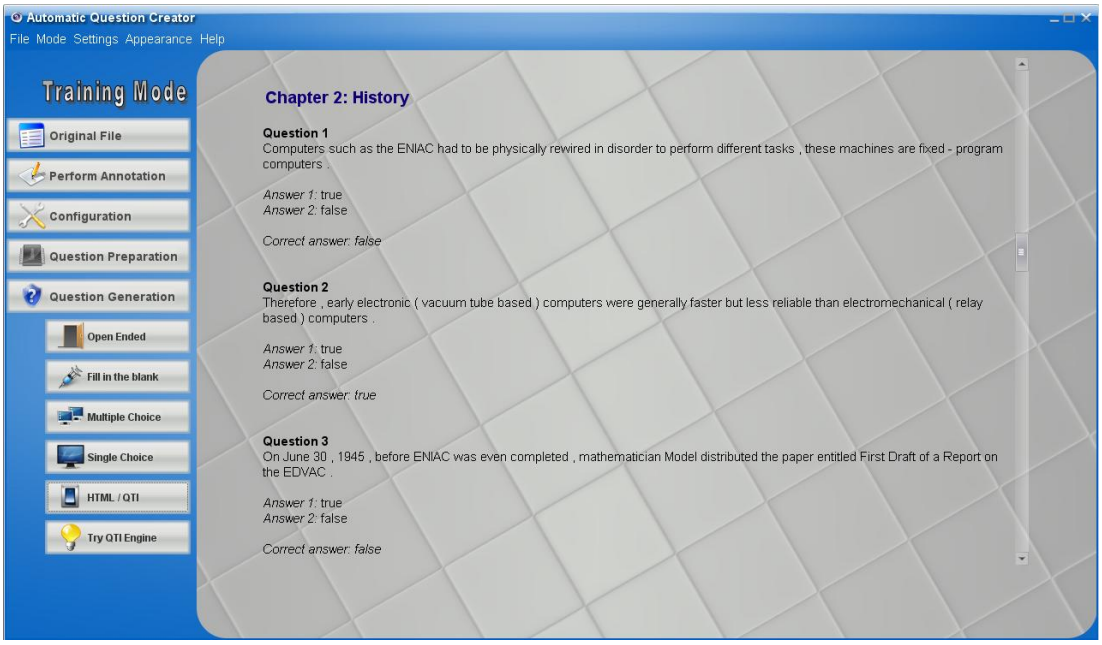


Abbildung 26: Single Choice HTML Darstellung

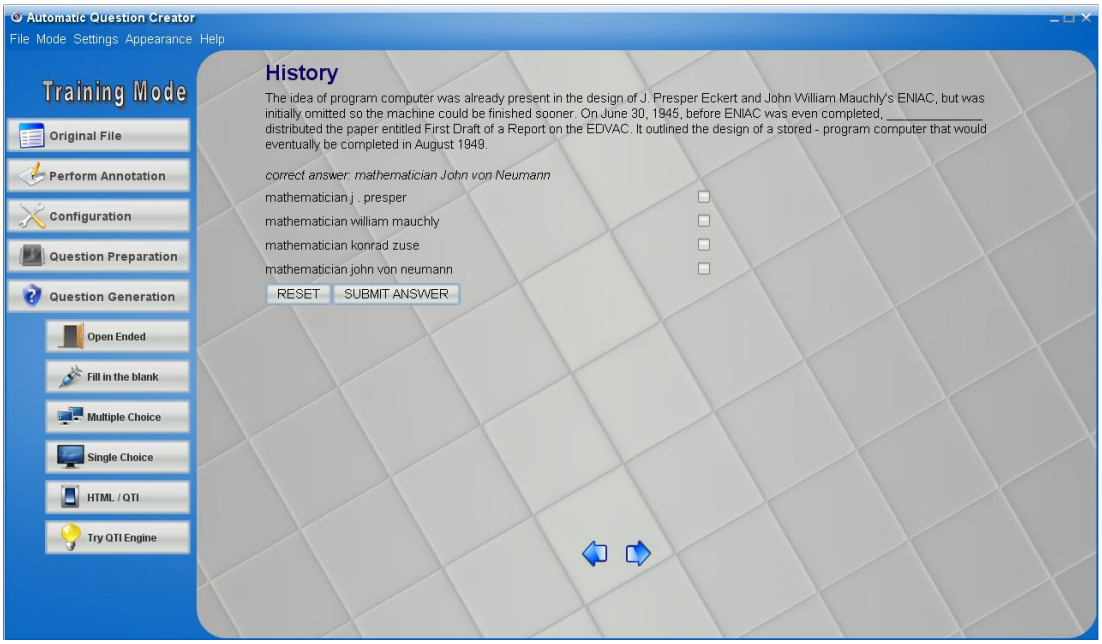


Abbildung 27: Multiple Choice QTI Darstellung

In Abbildung 26 wird ein Beispiel von Single Choice Fragen mit Ja / Nein Entscheidungen, in Abbildung 27 eine Multiple Choice Frage mit vier Antwortmöglichkeiten, wovon drei als Distraktoren fungieren, dargestellt.

6.5 Evaluierung

In diesem Subkapitel wird eine kleine Evaluierung vorgestellt, die das implementierte Programm bewerten und die generierten Fragen klassifizieren soll. Eingangs wird kurz die Vorgehensweise beschrieben und nachfolgend werden die aus der Evaluierung gewonnen Erkenntnisse aufgezeigt. Die Evaluierung wurde gemeinsam mit Weinhofer (2010) durchgeführt, die aufgeführten Ergebnisse beziehen sich jedoch nur auf den Teil der vorliegenden Arbeit.

6.5.1 Vorgehensweise

Die vier männlichen und die weibliche Testpersonen waren zwischen 24 und 30 Jahre alt. Vier davon befanden sich im Studium, wovon zwei Telematik, einer Betriebswirtschaftslehre und eine Betriebswirtschaftslehre, Psychologie und Pädagogik studierten. Ein weiterer Proband arbeitete als Softwareentwickler mit HTL Abschluss und 10-jähriger Berufserfahrung. Die Probanden wurden darum gebeten einen Teil eines englischsprachigen Textes über „Project Management for Construction“ von der MIT Open Course Ware (siehe MIT OCW, 2010) durchzulesen. Der Text um-

fasste etwa viereinhalb Seiten und war in drei Kapitel gegliedert. In weiterer Folge sollten die Versuchsteilnehmer die unten angeführten Punkte durchführen.

- Extrahieren Sie bitte die fünf wichtigsten Schlüsselphrasen und Konzepte jedes Kapitels!
- Bitte erstellen Sie zwei einfache Open Ended Aufgaben zu jedem Kapitel!
- Bitte erstellen Sie zwei einfache Fill In The Blank Aufgaben zu jedem Kapitel!
- Bitte erstellen Sie zwei einfache Single Choice zu jedem Kapitel!
- Bitte erstellen Sie zwei einfache Multiple Choice Aufgaben zu jedem Kapitel!

Im nächsten Schritt wurden die Testpersonen dazu angehalten, je Fragetyp zwei computergenerierte und zwei händisch erstellte Aufgaben für jedes Kapitel zu evaluieren. Je nach Fragetyp waren verschiedene Kriterien, die eine Abwandlung der Observation Matrix von Canella, Ciancimino und Campos (2010) darstellen, zu beurteilen. Die Observation Matrix kann im Falle des Automatic Question Creators nicht komplett übernommen werden, da die Terminologie, also die Wortwahl, und die Multidisciplinarity, welche die Verbindung mehrerer Themen ausdrückt, aufgrund der Programmstruktur nicht auszuwerten sind.

- **Pertinenz**: Beschreibt die Relevanz der Frage in Bezug auf die Thematik. (0...sehr gering; 100...sehr hoch)
- **Level**: Beurteilt den Schwierigkeitsgrad einer Frage.
  (0...sehr leicht; 100...sehr schwierig)
- **Konzept**: Beschreibt die Relevanz des Konzepts in Bezug auf die Thematik. (0...sehr gering; 100...sehr hoch)
- **Antwort**: Beschreibt die Qualität der berechneten Antwortregion bei Open Ended Fragen.
  (0...sehr gering; 100...sehr hoch)
- **Distraktoren**: Beschreibt die Qualität der berechneten Distraktoren bei Multiple Choice Fragen.
  (0...sehr gering; 100...sehr hoch)

Die Klassifikation der händisch erstellten Fragen hat den Sinn und Zweck zu überprüfen, inwieweit die Punktevergabe der Probanden bei realistischen Fragen erfolgt. Die Mittlung der obigen Kriterien soll Aufschluss über die Sinnhaftigkeit der einzelnen Fragetypen liefern. Abschließend wurde die Probanden allgemein über die Usabilty des Automatic Question Creators befragt und sollten sowohl negative als auch positive Erfahrungen im Umgang mit dem System schildern. Die Ergebnisse folgen im nächsten Unterpunkt dieses Kapitels.

6.5.2 Auswertung der Ergebnisse

In diesem Abschnitt werden die ausgewerteten, gemittelten Ergebnisse grafisch veranschaulicht und anschließend interpretiert. Die gezeigten Abbildungen zeigen dabei für die jeweiligen Fragetypen die Mittelwerte der zugehörigen evaluierten Kriterien, wobei händisch erstellte Fragen mit jenen die der Automatic Question Creator generiert hat, verglichen werden.

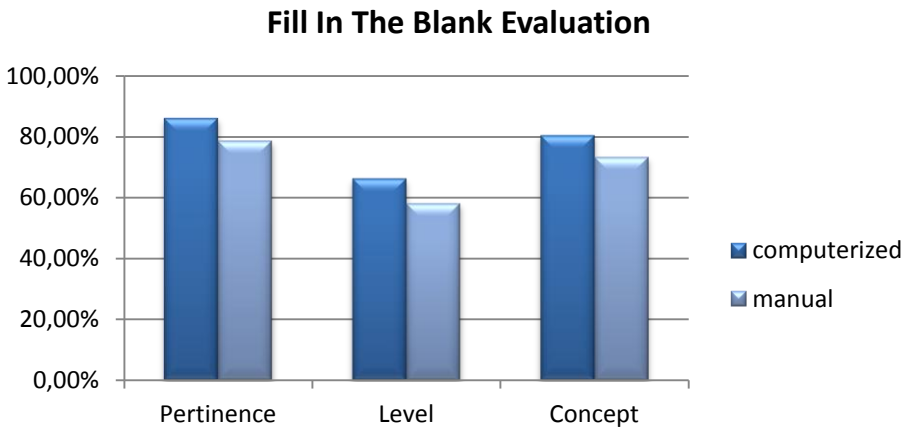


Abbildung 28: Fill In The Blank Evaluierung

Bei der Betrachtung der Ergebnisse der Fill In The Blank Evaluierung in Abbildung 28 wird ersichtlich, dass die Pertinenz und das ausgewählte Konzept, aufgrund dessen die Frage erzeugt wurde, bei computergestützt ermittelten Fragen die korrespondierenden händisch erstellten Fragen übertrifft. Der Schwierigkeitsgrad wurde bei automatisch generierten Fragen höher eingestuft was mit der Auswahl eines spezifischeren Konzepts begründet werden kann.

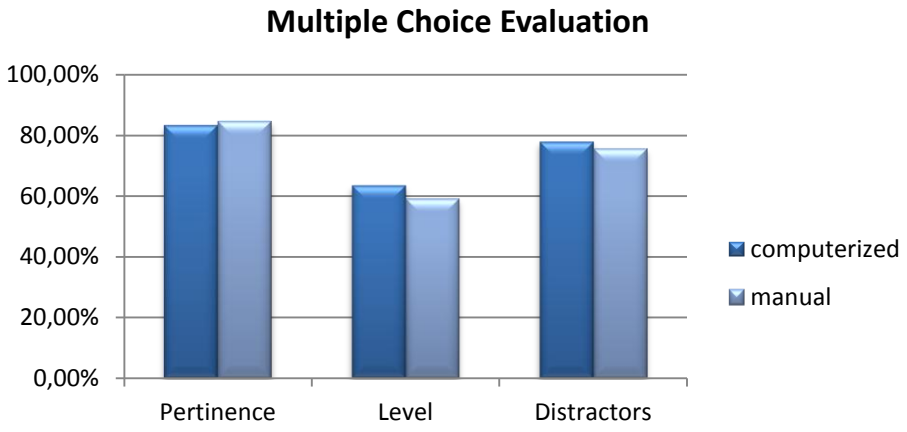


Abbildung 29: Multiple Choice Evaluierung

Abbildung 29 zeigt die durchschnittlich vergebenen Werte für die Pertinenz, den Schwierigkeitsgrad und die Qualität der Distraktoren bei Multiple Choice Fragen. Es zeigt sich, dass in jenem Fall die händisch erstellten Fragen von den Probanden als geringfügig relevanter eingestuft wurden, die Qualität der Distraktoren jedoch bei den computergenerierten Aufgaben besser beurteilt wurde. Der Schwierigkeitsgrad der vom Automatic Question Creator ermittelten Fragen wurde erneut höher eingestuft.

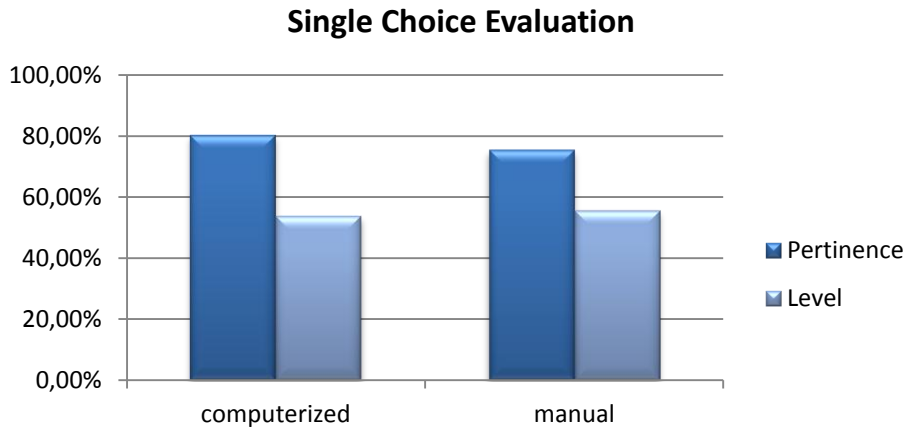


Abbildung 30: Single Choice Evaluierung

Der Vergleich der Punktevergabe der Testpersonen zwischen computererstellten und händisch erstellten Single Choice Aufgagen in Abbildung 30 verdeutlicht, dass der Schwierigkeitsgrad bei von Menschen erstellten Aufgaben in geringem Maße höher bewertet wurde.

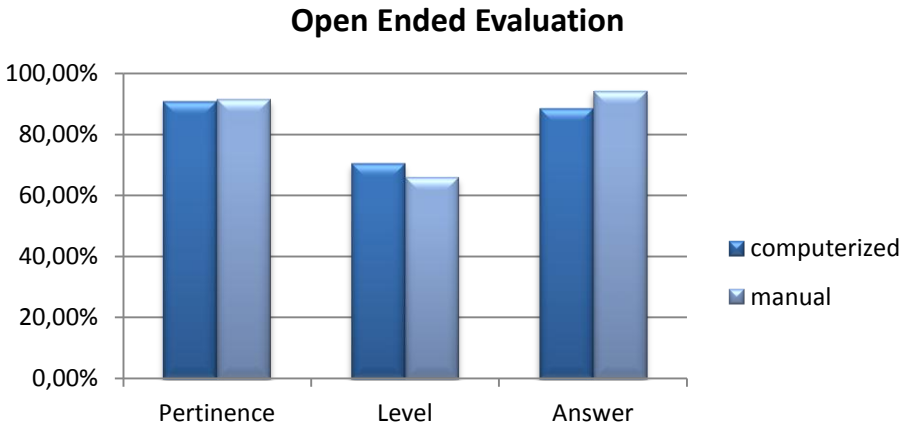


Abbildung 31: Open Ended Evaluierung

Die Evaluierung der Open Fragen in Abbildung 31 zeigt, dass die Pertinenz in beiden Fällen in etwa dieselbe ist, der Schwierigkeitsgrad bei computererstellten

Fragen geringfügig höher ist und die Berechnung der Antwortregion bei automatisiert erstellten Fragen schlechter beurteilt wurde.

Zusammenfassend kann, unter alleiniger Berücksichtigung der Werte, welche den Automatic Question Creator betreffen, gesagt werden, dass die Ergebnisse der durchschnittlichen Werte der Relevanz von Fragen aller Fragetypen eindeutig zeigen, dass jene mit Werten um die 80% als durchwegs relevant eingestuft wurden. Der Schwierigkeitsgrad wurde im Durchschnitt mit circa 60% angegeben und ist demnach von den Probanden als mittelmäßig schwer eingestuft worden. Die Qualität der Antwortregionen, welche klarerweise die richtigen Antworten beinhalten, die mit Werten von fast 90% eingestuft wurden legen jedoch dar, dass ein in der Länge nicht als störend empfundener Antwortbereich erzeugt wurde. Die Qualität der Distraktoren wurde ebenso mit circa 80% als gut befunden, die ausgewählten Konzepte erhielten ähnlich gute Werte. In Anbetracht der Tatsache, dass die Testteilnehmer für die Durchführung der gesamten Evaluierung im Schnitt fast zwei Stunden aufgewendet haben, wovon ein großer Teil für das Formulieren von und Suchen nach Fragestellungen herangezogen wurde, untermauern obige Ergebnisse den Erfolg des Automatic Question Creators und verdeutlichen, dass die erzeugten Fragestellungen durchaus brauchbar sind.

Die Auswertung der von den Probanden erzeugten Fragen der einzelnen Fragetypen hat gezeigt, dass extrem wenige untereinander übereinstimmende Aufgaben erstellt wurden. Dies betrifft sowohl die Auswahl der Konzepte, der Distraktoren als auch der Sätze für Fill In The Blank Fragen, was wiederum verdeutlicht, dass die Testpersonen sehr unterschiedliche Empfindungen über die Relevanz von der Wichtigkeit von Konzepten und Sätzen hatten.

Die allgemeine Befragung zur Handhabung und von Vor- sowie Nachteilen des Automatic Question Creators lieferte eindeutige Ergebnisse. Die relativ langen Ladezeiten während der Annotation und der Gewichtung wurden als störend empfunden. Die Probanden hatten keine Schwierigkeit mit dem Programm umzugehen und die Verarbeitungskette eigenständig durchzuführen. Die vielen Einstellungsmöglichkeiten wurden ebenso positiv hervorgehoben wie die Nachvollziehbarkeit der Gewichtung und der Konzeptauswahl.

6.6 Offene Erweiterungsmöglichkeiten

In diesem Subkapitel werden jene Komponenten und Erweiterungsvorschläge erklärt die in der Implementierung aus Zeitgründen keinen Platz gefunden haben, das Projekt jedoch einigermaßen vervollständigen würden.

6.6.1 Muster für Open Ended Fragen

Eine sinnvolle Erweiterung im Bezug auf die Erstellung von Open Ended Fragen wäre die Einführung von erweiterten Mustern. Einerseits könnten Entwurfsmuster für alle in GATE verfügbaren Annotationstypen implementiert werden, andererseits wäre es jedoch auch möglich, die Fragen anstatt einer alleinigen Abhängigkeit von der Fragephrase auch in Abhängigkeit des gesamten Satzes zu konstruieren. Jene Änderung würde sehr komplexe Muster erfordern, da mit steigender Satzlänge die Anzahl von enthaltenen speziellen Annotationen ebenso zunimmt. Es müssten also Querverbindungen zwischen verschiedenen Konzepten hergestellt werden, die eine komplexe Planung erfordern.

6.6.2 WordNet Distraktorenverbesserung

Die mit Hilfe von WordNet extrahierten Distraktoren zu einem Wort beziehungsweise einer Phrase können verbessert werden, indem neben den einfachen Relationen die in WordNet gespeicherten Kontextbeschreibungen mit dem Kontext des Ausgangswortes verglichen werden und nur jene Relationen berücksichtigt werden, deren Kontext mit dem Kontext im Text eines definierten Bezug oder zumindest einen hohen Ähnlichkeitsgrad aufweisen. Diese Berechnung ist sicherlich relativ diffizil und es könnte daraus das Problem resultieren, dass zu wenige alternative Konzepte zur Verfügung stehen. Trotz alledem könnte die Qualität der Distraktoren dadurch erheblich gesteigert werden.

Zusätzlich kann, wie die Autoren Brown, Frishkoff und Eskenazi, (2005) vorschlagen, die Auftrittshäufigkeit eines Wortes im allgemeinen Sprachgebrauch berücksichtigt werden, indem potentiell nur jene Wörter aus WordNet ausgewählt werden, die eine ähnliche Auftrittswahrscheinlichkeit besitzen wie das erfragte Wort, was die Qualität der Distraktoren ebenso verbessern kann.

6.6.3 Klassifikation der generierten Aufgaben

Da der vorgestellte Ansatz beziehungsweise das implementierte System und die bisherigen Forschungsansätze gezeigt haben, dass immer eine gewisse Anzahl an automatisiert erstellten Aufgaben entweder irrelevant oder aufgrund der Aufgabenstellung unbrauchbar sind, wäre es wünschenswert, Methoden zu entwickeln, welche basierend auf händischer Klassifikation der Aufgaben eine maschinelle Klassifikation umsetzen.

Durch Analysen der Sätze und Schlüsselterme welche unzureichende Aufgaben hervorgerufen haben, könnten durch die Erkenntnisse hinsichtlich der POS Tags, der speziellen Annotationen und der Satzstruktur Klassifikatoren entwickelt werden, welche jene Aufgaben herausfiltern.

6.6.4 Fragenauswertung

Die Art und Weise der automatischen Bewertung einer Frage hängt natürlich von dem Fragetyp ab. Nachfolgend wird kurz erläutert, inwieweit eine automatische Bewertung sinnvoll ist und welche Probleme sich daraus ergeben können.

Die einfachste Art der Benotung einer Antwort findet sich bei Single Choice Fragen, also jenen Fragen bei denen aus zwei vorgegebenen Antwortmöglichkeiten zu wählen ist, wieder. Eine Antwort kann entweder richtig oder falsch sein und es sind keine weiteren Berechnungen nötig, die erreichte Punkteanzahl ist im QTI Schema eingetragen, wobei natürlich auch negative Punkte vergeben werden können.

Bei Multiple Choice Fragen ist die Situation ähnlich. Grundsätzlich können bei dieser Art von Fragen keine bis alle Antwortmöglichkeiten richtig sein. Die Punktevergabe richtet sich wieder nach den im QTI Standard vergebenen Punkten. Analog dazu kann bei Lückentexten vorgegangen werden, sofern die Antwortmöglichkeiten in Form von Multiple Choice Fragen angegeben sind. Sollte die Lücke selbstständig auszufüllen sein wird es diffiziler, da ohne Hilfsmittel keine eindeutige Auswertung möglich ist. Wird das erfragte Wort nicht eins zu eins wiedergegeben, können zweierlei Probleme auftreten. Einerseits könnte das Wort falsch geschrieben sein, wobei die Nutzung von N-Grammen Abhilfe schaffen kann, indem festgelegt wird, dass beispielsweise nur eine gewisse Anzahl an Tri-Grammen übereinstimmen muss. Bei der Beantwortung der Frage durch ein anderes Wort als durch das Erfragte kann WordNet dazu genutzt werden, um die Ähnlichkeit der Wörter zu ermitteln. Das beste Resultat würde ein Synonym erzielen, je nach Bedarf und Wortart könnte auch eine meronyme, hyponyme, holonyme oder troponyme Relation ausgewertet werden und die im QTI Standard festgelegte Punktevergabe mit einem Faktor gewichtet, in diesem Fall abgeschwächt, werden.

Am kompliziertesten ist die Auswertung von Antworten auf offene Fragen, da der verwendete Wortschatz vom Vokabular im Text abweichen kann. Im Wesentlichen erfolgt die Auswertung am besten durch die Überführung der Sätze in Vektoren und die anschließende Bestimmung des Ähnlichkeitsmaßes. Ein Beispiel für diese Vorgehensweise ist die Überprüfung mittels des Vector Space Models (siehe auch Kapitel 2.1.4.3). Es werden zwei Vektoren, in diesem Fall aus der Referenzantwort und der tatsächlichen Antwort, erstellt und deren Distanz berechnet. Verbessert können die Resultate dieses einfachen Modells werden, indem das Critical Sentence Vector Model von Li, Wong, Yuan, Li und Xia (2005) eingesetzt wird. Bei diesem werden die SVO Struktur und die längsten gemeinsamen Wortfolgen der beiden Textteile gesondert berücksichtigt. Die Punktevergabe hängt erneut von den Bedürfnissen, den Anforderungen und dem Themengebiet ab, der Meinung des Autors nach ist die Auswertung der SVO Struktur einer Antwort die sinnvollste, da darin die meiste Information beinhaltet ist.

Weitere mögliche Beurteilungskriterien sind die Precision, also der Anteil der relevanten Wörter im Vergleich zu allen angegeben Wörtern, wobei vorwiegend Hauptwörter ausgewertet werden sollten, und der Recall, also das Verhältnis von der Anzahl der angegebenen richtigen Wörter und aller richtigen Wörter. In bestimmten Situationen können jene beiden Bewertungskriterien jedoch auch auf Verben und bei sehr detaillierten Informationen auch auf Adjektive angewendet werden.

Abschließend wird erneut die Abhängigkeit von WordNet erwähnt, da Wörter, die nicht in WordNet erfasst sind nicht ausgewertet werden können, wenn der Wortschatz von Antworten nicht mit dem im Text übereinstimmt.

6.6.5 Feedback

Geeignetes Feedback ist für den Nutzer essentiell, da ohne jenes das Lernen erschwert wird und nicht verstanden werden kann, welche Schwierigkeiten noch zu bewältigen sind. Zusätzlich wirkt Feedback motivierend und reduziert die Zeit des Lernens, da gezielter gelernt werden kann.

Automatisch generiertes Feedback zu einem Thema oder Begriff kann im Wesentlichen auf drei verschiedene Arten generiert werden. Einerseits kann dem Nutzer der Einfachheit halber eine vorab definierte Referenzantwort angezeigt werden, andererseits aber auch einfach jener Abschnitt dargelegt werden, welcher das Erfragte beinhaltet. Diese Techniken mögen zwar ausreichend sein, sind der Meinung des Autors nach aber zu oberflächlich. Die dritte und bessere Methode zum generieren von automatischem Feedback ist eine automatische Zusammenfassung der angesprochenen Textpassage. Es wurden in Kapitel 3 einige Ansätze vorgestellt, die Konzepte beinhalten um dies durchzuführen, da dies jedoch den Fokus der Aufgabenstellung verlässt, wird nicht näher darauf eingegangen wie im Detail eine solche durchzuführen ist.

Wie bereits in Kapitel 2.2.1 erwähnt kann laut Roberts (2006) das Konzept Peer-Assessment, also die gegenseitige Beurteilung von Lernenden beziehungsweise das gegenseitige geben von Feedback oftmals besser aufgenommen werden und sollte deshalb nicht außer Acht gelassen werden. Ein optimales System bietet demnach die Möglichkeit, neben der automatischen Bewertung und automatischem Feedback nach gewissen Regeln alle Teilnehmer in diesen Prozess mit einzubeziehen.

6.7 Zusammenfassung

In diesem Kapitel wurde eingangs das konzeptionelle Design des Automatic Question Creators vorgestellt und aufgezeigt, dass im Wesentlichen eine geeignete Vorverarbeitung, eine Konzeptauswahl und die eigentliche Fragengenerierung erfolgen. Es war auch ersichtlich, dass die externen Frameworks von GATE, WordNet, Extrak4Me und TextTiling in die Implementierung eingebunden sind.

In der Beschreibung der Implementierung wurde zu Beginn die Vorverarbeitung, bestehend aus einer Annotation und einem Überführen in eine spezifische Datenstruktur beschrieben. Darauf folgend wurde gezeigt wie mittels des ExtraK4Me Algorithmus wichtige Schlüsselphrasen identifiziert und mit den Ergebnissen der Konzeptextraktion von Weinhofer (2010) kombiniert wurden. Zusätzlich zeigte sich, dass das Question Preprocessing Modul es dem User erlaubt, die bis dahin kalkulierten Phrasen zu manipulieren und in den Prozess der Aufgabenerstellung einzugreifen.

Anschließend wurde erklärt wie die eigentliche Aufgabenerstellung abläuft und wie im Detail die Open Ended, Fill In The Blank, Single Choice und Multiple Choice Fragen sowie die Distraktoren konstruiert werden. Abschließend wurde aufgezeigt das die ermittelten Fragen im QTI Standard gespeichert werden, um eine spätere Weiterverarbeitung zu gewährleisten.

Im Anschluss an die Details zur Implementierung wurden die dabei aufgetretenen Probleme dargelegt. Im Wesentlichen sind dies Schwachstellen bezüglich WordNet und GATE, auf die kein Einfluss genommen werden kann. Die hohe Laufzeit ergibt sich aus dem Aufbau des QTI Standards, da sehr viele Dateien erzeugt werden müssen.

Im nächsten Abschnitt dieses Kapitels wurde die Sichtweise des Users dargelegt, indem wesentliche Screenshots von der sequentiellen Vorgehensweise vom annotierten Dokument bis hin zur Fragenanzeige im HTML Format und in der QTI Repräsentation gezeigt und kurze Erklärungen dazu geliefert wurden.

Abschließend wurden offene Erweiterungsmöglichkeiten dargelegt, welche im Wesentlichen die Verbesserung der Distraktorenauswahl, die Reformulierung von Open Ended Fragen, die Klassifikation der automatisiert erstellten Aufgaben, eine automatische Auswertung von Antworten in einer Assessment Situation sowie die Generierung von Feedback auf gegebene Antworten sind.

Im nächsten Kapitel werden die vom Autor dieser Arbeit erfahrenen Lektionen beschrieben und gezeigt, welche Schwierigkeiten bei der Erstellung dieser Arbeit aufgetreten sind.

7 *Lessons Learned*

Bei der Erstellung dieser Arbeit, die intensive Recherchen und eine aufwendige Implementierung beinhaltet, hat der Autor dieser Arbeit vor allem erfahren, dass wissenschaftliches Arbeiten in diesem Umfang sehr hartes Arbeiten bedeutet. Es ist besonders wichtig, sich vorab ein Konzept zu erstellen und ein derartiges Projekt nicht ziellos anzugehen. Wesentlich ist auch das konsequente Verfolgen eines Fadens sowohl in Bezug auf das Recherchieren, das Schreiben als auch das Implementieren, um sich nicht in nebensächliche Details zu verrennen.

Zu Beginn der Arbeit war es sehr schwierig sich einen Überblick über die grundlegenden Theorien, das Natural Language Processing und Assessment, zu verschaffen, da vor allem ersteres ein sehr umfangreiches Gebiet darstellt. Es musste herausgearbeitet werden, welche Teilgebiete am relevantesten für die Aufgabenstellung der automatisierten Aufgabenerstellung sind, da es zu dieser Thematik im konkreten sehr wenig explorative Studien und brauchbare Ansätze gibt. In Bezug auf die aktuellen Forschungsansätze ist anzumerken, dass aufgrund der Vielzahl derer nur ausgewählte Artikel gelesen und behandelt werden konnten und es anfangs sehr schwierig war, schnell zwischen guten und schlechteren Publikationen zu unterscheiden. Es trat hierbei auch das Problem auf, dass viele Papers zwar offensichtlich sehr konkrete Fragestellungen behandeln, die Art und Weise der realen Umsetzung der Problemlösung jedoch oftmals relativ schwammig und nicht nachvollziehbar erfolgt.

Die Implementierung des Automatic Question Creator, für die die Erkenntnisse aus der Entwicklung eines, gemeinsam mit Weinhofer Joachim entwickelten, kleinen Prototypen genutzt werden konnten, gestaltete sich ebenfalls teilweise sehr mühsam, wobei weniger die programmiertechnischen Fähigkeiten sondern vielmehr die breit gefächerten Anforderungen und die Vielzahl an integrierten Ansätzen den enormen Zeitaufwand verursachten. Zusätzlich erwies sich die oftmals kurz gehaltene Beschreibung der verwendeten Frameworks als Hindernis, da die Zeiteinschätzung der Einbindung einige Male unterschätzt wurde. Eine weitere Problematik, die im Zuge der Implementierung zum Vorschein trat, waren die teilweisen falschen beziehungsweise unzufrieden stellenden Ergebnisse von WordNet und GATE.

Trotz all der geschilderten Hindernisse war der Autor der Arbeit stets motiviert, was vor allem durch die Neuartigkeit des implementierten Systems und die damit verbundene Handlungsfreiheit bedingt war. Im Grunde genommen war es auch eine sehr große Herausforderung ein Ende zu finden und nicht alle Ideen tatsächlich umzusetzen.

8 *Zusammenfassung*

Das Ziel dieser Masterarbeit war die Implementierung eines Tools, welches in der Lage ist, basierend auf vorab extrahierten Konzepten und den damit einhergehenden Analysen beziehungsweise den daraus gewonnenen Erkenntnissen, automatisiert Aufgaben zu erstellen. Die konzeptionelle Vorgehensweise sollte dabei so gewählt werden, dass ein möglichst verallgemeinerbarer Ansatz implementiert wird und die natürlich sprachlichen Input Dokumente keinen Einschränkungen unterliegen.

Um dieser Aufgabenstellung gerecht zu werden wurde zu Beginn der Arbeit eine Recherche über die Thematik des Natural Language Processing angestellt. Es konnte dabei aufgezeigt werden, dass die komplexe Problemstellung syntaktische, pragmatische, semantische und statistische Analysen erfordert, damit die benötigten Informationen für die Konzeptextraktion und Aufgabenerstellung erlangt werden können. Zusätzlich wurden Assessmentsysteme in Bezug auf deren Zweck und Methodik untersucht und als wesentliche Bestandteile der Anforderungen an ein derartiges System die Flexibilität, der Einsatz von adäquaten technischen Mitteln, Privatsphäre, Austauschbarkeit, Modularität, die Einhaltung von Standards und eine leicht verständliche Benutzeroberfläche sind.

Neben der Aneignung der abstrakten theoretischen Grundlagen hat der Autor dieser Arbeit versucht, sich einen Überblick über den aktuellen Forschungsstand von Konzeptextraktion, automatischer Aufgabenerstellung und Assessment-Systemen zu verschaffen. Dabei wurde ersichtlich, dass es vor Allem in Bezug auf die automatische Aufgabenerstellung Defizite vorhanden sind und diese Disziplin vergleichsweise unerforscht ist. Darüber hinaus wurde ersichtlich, dass bei der Ermittlung der Schlüsselkonzepte meistens eine thematische Abhängigkeit gegeben ist, was vorwiegend durch die Trainingsmengen bei maschinellem Lernen bedingt ist. Ein beinahe immer geltendes und auch Anwendung findendes Konzept ist jenes der Worthäufigkeiten, selbiges gilt für den Einsatz von Tools wie WordNet und GATE.

Darauf folgend wurde sowohl in Bezug auf die Anforderungen als auch auf das konzeptionelle Design ein umfassender Ansatz vorgestellt, der den Gesamtprozess der textuellen Vorverarbeitung, der Überführung in eine interne Datenstruktur, der Konzeptextraktion, der Konzeptauswahl, der Aufgabenerstellung, der Aufgabenauswertung und der Generation geeigneten Feedbacks beinhaltet. Es zeigten sich vor allem der Wunsch nach einer allgemeinen Lösung, der Unterstützung von Open Ended, Fill In The Blank, Single Choice und Multiple Choice Aufgaben, standardisierten Formaten wie IMS QTI, sowie einer geeigneten und genügend Einstellungsmöglichkeiten bietenden Benutzeroberfläche.

Weiters wurden die in der Implementierung verwendeten Java Tools und Frameworks dargelegt und herausgearbeitet, dass GATE für eine Annotation des Textes angewandt werden kann, da ein übersichtliches Annotationsschema geboten werden, eine Vielzahl an Annotationstypen unterstützt werden und ein Fülle an Plugins zu Verfügung stehen. Darüber hinaus wurden die Funktionalität und der Aufbau der elektronischen, lexikalischen Datenbank WordNet, das JQTI Framework beziehungsweise der QTI Standard, der TextTiling Algorithmus, XtraK4Me und Synthetica vorgestellt, sowie aufgezeigt, warum jene Tools in der Implementierung eingesetzt werden.

Anschließend wurden das konzeptionelle Design und konkrete Implementierungsdetails des Automatic Question Creators vorgestellt. Es zeigte sich, dass aufbauend auf die von Weinhofer (2010) und Extrak4Me durchgeführte Konzeptextraktion, das Question Preprocessing Modul folgt, welches anhand von Gewichtsberechnungen die zur Verfügung stehenden Konzepte reiht, dem User präsentiert und diesen durch Ab- und Auswahl von ausgewählten und fehlenden Konzepten in den Prozess eingreifen lässt. Das Question Generation Modul bietet im Wesentlichen die Möglichkeit, Aufgaben zu den vier Fragetypen Open Ended, Fill In The Blank, Single Choice und Multiple Choice zu erstellen, geeignete Distraktoren zu ermitteln und Referenzantworten zu kalkulieren. Die Aufgabenerstellung ist dabei von vorab durchgeführten Textanalysen abhängig, da je nach Fragetyp Wortgruppen, spezielle Annotationstypen, Distraktoren und Antonyme ermittelt werden müssen.

Bei Fill In The Blank Fragen werden Schlüsselkonzepte ausgeblendet, bei Single Choice Fragen durch Antonyme oder spezielle Annotationen selben Types ausgetauscht und bei Open Ended Fragen anhand von Mustern in Abhängigkeit des Kontexts erfragt. Zusätzlich werden mittels des TextTiling Algorithmus jene Textgrenzen gesucht, welche das Schlüsselkonzept beinhalten und der enthaltene Text als Referenzantwort gespeichert. Multiple Choice Fragen sind analog zu Lückentexten aufgebaut und bieten dem Nutzer vier potentielle Antwortmöglichkeiten, wobei drei anhand von WordNet und so genannten coordinate terms oder durch Berücksichtigung von speziellen Annotationen berechnet werden. Die erzeugten Aufgaben werden im QTI Standard umgesetzt und können somit problemlos in andere Tools oder Assessmentsysteme integriert werden.

Durch die Präsentation der Sichtweise des Benutzers wurde ein grober Überblick über die Funktionalität, die Visualisierung und die Benutzerinteraktion des Automatic Question Creators gegeben und durch einige Screenshots verdeutlicht. Darüber hinaus wurden offene Erweiterungsmöglichkeiten, wie die Verbesserung der Fragenformulierung, der Distraktorenauswahl, der Fragenauswertung und dem Generieren von Feedback aufgezeigt und mögliche Herangehensweisen an jene Problemstellungen erklärt.

Anschließend wurde von fünf unabhängigen Testpersonen eine kleine Evaluierung von einigen, vom implementierten System erstellten, Fragen durchgeführt. Es hat sich gezeigt, dass die Relevanz, der Schwierigkeitsgrad, das ausgewählte Konzept, die Referenzantwort und die Distraktorenqualität der einzelnen generierten Fragen von allen Nutzern sehr positiv beurteilt wurden. Zusätzlich konnte im Rahmen dieser Untersuchung aufgezeigt werden, dass die generierten Fragen im Vergleich zu händisch erstellten Fragen ungefähr gleich gut bewertet wurden. Darüber hinaus wurde herausgefunden, dass beinahe jeder Versuchteilnehmer andere Konzepte und Sätze als wichtig empfunden hat und nur sehr wenige Zusammenhänge bei Fragen, welche die Nutzer zu einem Testdokument erstellen mussten, hervorgetreten sind.

Abschließend kann vom Autor dieser Masterarbeit zusammengefasst werden, dass die Implementierung, welche nach den Erkenntnissen der intensiven Recherchen entstanden ist, den Ergebnissen nach ein voller Erfolg ist. Die aufgezeigten Verbesserungsmöglichkeiten könnten die bestehenden Schwächen ausbessern und eine Weiterentwicklung jener Tools, welche in der Implementierung integriert sind, ebenso positive Auswirkungen auf die erzielten Resultate haben.

9 *Literaturverzeichnis*

Augst, G. (1975). Lexikon zur Wortbildung Morpheminventar A-G. In *Forschungsberichte des Instituts für deutsche Sprache* (Vol. 24.1). Tübingen: TBL Verlag Gunter Narr.

Baader, F., Horrocks, I., & Sattler, U. (2004). Description Logics. In Staab, S., & Studer, R. (Hrsg.) *Handbook on ontologies* (Seiten 3 - 28). Heidelberg: Springer Verlag.

Baek, S., Cho, M., & Kim, P. (2005). Matchings Colors with KANSEI Vocabulary Using Similarity Measure Based on WordNet. In Gervasi, O., Gavrilova, M. L., Kumar, V., Laganá, A., Lee, H. P. Mun, Y., Taniar, D., & Tan, C. J. K. (Hrsg.), *Computational Science and Its Applications – ICCSA 2005, International Conference, Singapore, May 2005, Proceedings, Part 1* (Seiten 37 - 45). Berlin, Heidelberg: Springer Verlag.

Baeza-Yates, R. (2004). Challenges in the Interaction in Information Retrieval and Natural Language Processing. In Gelbukh, A. (Hrsg.): *Computational Linguistics and Intelligent Text Processing, 5th International Conference, CICLing 2004, Seoul, Korea, February 2004, Proceedings* (Seiten 445 – 456). Berlin, Heidelberg: Springer Verlag.

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

Barzilay, R., & Elhadad, M. (1997). Using Lexical Chains for Text Summarization. In Mani, I., & Maybury, M. (Hrsg.), *Proceedings of the ACL/EACL 1997 Workshop on Intelligent Scaleable Text Summarization, Madrid, Spain, July 1997* (Seiten 10 - 17). Morristown: Association for Computational Linguistics.

Berlak, H., Newmann, F., Adams, E., Archbald, D., Burgess, T., Raven, J., & Romberg, T. A. (1992). *Toward a new science of educational Testing & Assessment*. New York: State University of New York.

Blei, D. M., & Moreno, P. J. (2001) Topic Segmentation with an Aspect Hidden Markov Model. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (Seiten 343 - 348). New York: Association for Computing Machinery

Bin, L, Jun, L., Jian-Min, Y., & Qiao-Ming, Z. (2008). Automatic Essay Scoring Using the KNN Algorithm. In *Proceedings: International Conference on Computer Sci-*

ence and Software Engineering, CSSE 2008, 1 (Seiten 735 - 738). Washington DC: IEEE Computer Society.

Brown, J. C., Frishkoff, G. A., & Eskenazi, M. (2005). Automatic Question Generation for Vocabulary Assessment. In *Proceedings of the Human Language Technology Conference on Empirical Methods in Natural Language Processing: 6 – 8 October 2005, Vancouver, British Columbia, Canada* (Seiten 819 - 826). Morristown: Association for Computational Linguistics.

Brückner, T. (2001). Textklassifikation. In Carstensen, K.-U. (Hrsg.), Anwendungen. In Carstensen, K.-U., Ebert, C. ; Endriss, C., Jekat, S., Klabunde, R., & Langer, H. (Hrsg.), *Computerlinguistik und Sprachtechnologie: Eine Einführung* (Seiten 442 – 447). Heidelberg: Spektrum Akademischer Verlag.

Chen, R. (2003). English Inversion: A Ground-before-figure construction. Berlin: Mouton de Gruyter.

Chen, C.-Y., Liou, H.-C., & Chang, J. S. (2006). FAST: An Automatic Generation System for Grammar Tests. In *Annual Meeting of the ACL archive: Proceedings of the COLING/ACL on Interactive presentation sessions* (Seiten 1 - 4). Morristown: Association for Computational Linguistics.

Canella, S., Ciancimino, E., & Campos, M. L. (2010). Mixed e-Assessment: an application of the student generated question technique. *IEEE Engineering 2010. Engineering Education Conference*.

Coniam, D. (1997). A Preliminary Inquiry Into Using Corpus Word Frequency Data in the Automatic Generation of English Language Cloze Tests. In *Calico Journal Vol. 14*, (2 – 4) (Seiten 15 - 34).

Cristea, D., Postolache, O., & Pistol, I. (2006). Summarisation Through Discourse Structure. In Gelbukh, A. (Hrsg.): *Computational Linguistics and Intelligent Text Processing, 6th International Conference, CICLing 2005, Mexiko City, Mexiko, February 2005, Proceedings* (Seiten 632 – 644). Berlin, Heidelberg: Springer Verlag.

Cunningham, H., Maynard, D., Bontcheva, K., & Tablan, V. (2002). GATE: an Architecture for Development of Robust HLT Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics ACL´02* (Seiten 168 - 175).

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Ursu, C., Dimitrov, M., Dowman, M., Aswani, N., Roberts, I., Li, Y., Shafirin, A., & Funk, A. (2009). De-

veloping Language Processing Components with GATE Version 5 (a User Guide). URL - http://gate.ac.uk/sale/tao/tao.pdf (Zugriffsdatum 11.02.2010)

Delozanne, E., Prévit, D., Grugeon, B., & Chenevotot, F. (2008). Automatic Multi-criteria Assessment of Open-Ended Questions: A Case Study in School Algebra. In Woolf, B. P., Aimeur, E., Nkambou, R., & Lajoie, S. (Hrsg.), *Intelligent Tutoring Systems, 9th International Conference, ITS 2008, Montreal, Canada, June 2008, Proceedings* (Seiten 101 - 110). Berlin, Heidelberg: Springer Verlag.

De Busser, R. (2006). Information Extraction and Information Theory. In Moens, N.-F. (Hrsg.), Information *Extraction: Algorithms and Prospects in a Retrieval Context* (Seiten 1- 22). Dordrecht: Springer Verlag.

Dikli, S. (2006). An Overview of Automated Scoring of Essays. In Russel, M. (Hrsg.), *The Journal of Technology, Learning and Assessment, 5* (1).

Endres-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 456 – 461). Berlin, Heidelberg: Spektrum Akademischer Verlag.

Fellbaum, C. (1998). A Semantic Network of English Verbs. In Fellbaum, C. (Hrsg.), *WordNet: an electronic lexical database* (Seiten 69 - 104). Massachusetts: MIT Press.

Fischer, S., & Steinmetz, R. (2000). Automatic Creation of Exercises in Adaptive Hypermedia Learning Systems. In: *Proceedings of the eleventh ACM on Hypertext and hypermedia* (Seiten 49 -55). San Antonio: ACM.

Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C., & Nevill-Manning, C. G. (1999). Domain-specific Keyphrase Extraction. In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence IJCAI (Seiten 668 - 673). Morgan Kaufmann Publishers.

Gansel, C., & Jürgens, F. (2007). *Textlinguistik*. Göttingen: Vandenhoeck & Ruprecht.

GATE (2010a). GATE: general architecture for text engineering. URL - http://gate.ac.uk/ (Zugriffsdatum 10.02.2010)

GATE (2010b). *AKT: Advanced Knowledge Technologies: ANNIE – Open Source Information Extraction from the University of Sheffield*. URL - http://www.aktors.org/technologies/annie/ (Zugriffsdatum 25.02.2010)

Giménez, J., & Márquez, L. (2004). Fast and accurate part-of-speech tagging: The SVM approach revisited. In Nicolov N., Bontcheva, K., Angelova, G., & Mitkov, R. (Hrsg.), *Recent Advances in Natural Language Processing III* (Seiten 153 – 162). Amsterdam: John Benjamins Publishing.

Granger, S. (2002). A Bird´s-eye view of learner corpus research. In Granger, S., Hung, J., & Petch-Tyson, S. (Hrsg.), *Computer Learner Corpora, Second Language Acquisition and Foreign Language Teaching* (Seiten 3 – 36). Amsterdam: John Benjamins Publishing.

Grefenstette, G., & Tapanainen, P. (1994). What is a word, What is a sentence? Problems of Tokenization. In Complex ’94, *Proceedings of the 3rd International Conference on Computational Lexicography* (Seiten 79–87). Research Institute for Linguistics, Hungarian Academy of Sciences.

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

Gütl, C. (2008a). Automatic Limited-Choice and Completion Test Creation, Assessment and Feedback in modern Learning Processes. In *LRN Conference 2008, Guatemala, February 12th – 16th, 2008*.

Gütl, C. (2008b). Moving towards a Fully Automatic Knowledge Assessment Tool. In *iJET International Journal of Emerging Technologies in Science, 3* (1) (o.S.).

Gütl, C., & AL-Smadi, M. (2008). Past, Presence and Future of e-Assessment: Towards a Flexible e-Assessment System. In *Conference of Interactive Computer Aided Learning, Villach*.

Gütl, C., AL-Smadi, M., & Kappe, F. (2009). PASS: Peer ASSessment Approach for Modern Learning Settings. In Spaniol, M., Li, Q., Klamma, R., & Lau, R. W. H. (Hrsg.), *Advances in Web Based Learning – ICWL 2009: 8th International Conference Aachen, Germany, August, 2009, Proceedings* (Seiten 44 – 47), Heidelberg: Springer Verlag.

Hall, T. A. (2000). *Phonologie: Eine Einführung*. Berlin: de Gruyter Studienbuch.

Hammouda, K. M., Matute, D. N., & Kamel, M. S. (2005). CorePhrase: Keyphrase Extraction for Document Clustering. In Perner, P., & Imiya, A. (Hrsg.), *Machine Learning and Data Mining in Pattern Recognition, 4th International Conference, MLDM 2005, Leipzig, Germany, July 2005, Proceedings* (Seiten 265 - 274). Berlin, Heidelberg: Springer Verlag.

Hassan, S., Mihalcea, R., & Banea, C. (2007). Random-Walk Term Weighting for Improved Text Classification. In *Semantic Computing, ICSC 2007* (Seiten 242 - 249). Institute of Electrical and Electronics Engineers (IEEE).

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2. Auflage). New York: Springer Science+Business Media.

Haton, J.-P. (1987). Knowledge-Based and Expert Systems in Understanding Problems. In Haton, J.-P. (Hrsg.), *Fundamentals in Computer Understanding: Speech and Vision* (Seiten 1 – 22). Cambridge: Cambridge University Press.

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

Hearst, M. A. (1997). Texttiling: segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, 22 (1) (Seiten 33 - 64).

Hearst, M. A. (1998). Automated Discovery of WordNet Relations. In Fellbaum, C. (Hrsg.), *WordNet: an electronic lexical database* (Seiten 133 - 151). Massachusetts: MIT Press.

Hearst, M. A. (2000). The debate on automatic essay grading. *IEEE Intelligent Systems, 15* (5) (Seiten 22 - 37).

Hess, M., & Clematide, S. (2008). *Chunk Parsing*. November 2008. – URL http://kitt.ifi.uzh.ch/clab/chunking/chunk_parsing.pdf (Zugriffsdatum: 15. Jänner 2010)

Horrocks, I., Patel-Schneider, P. F., & van Harmelen, F. (2003). From SHIQ, and RDF to OWL: The Making of a Web Ontology Language. In *Journal of Web Semantics, 1* (1) (Seiten 7 – 26).

Hoshino, A., & Nakagawa, H. (2005). A real-time multiple-choice question generation for language testing: a preliminary study. In *Proceedings of the second workshop on Building Educational Applications Using NLP* (Seiten 17 - 20). Morristown: Association for Computational Linguistics.

Jackson, P., & Moulinier, I. (2007). *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization* (2. überarbeitete Auflage). Amsterdam: John Benjamins Publishing.

Joosten-ten Brinke, D., Gorissen, P., & Latour, I. (2005). Integrating Assessment into E-learning Courses. In Koper, R., & Tattersall, C. (Hrsg.), *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training* (Seiten 185 - 202). Berlin, Heidelberg: Springer Verlag.

Justeson, J. S., & Katz, S. L. (1996). Technical terminology: some linguistic properties and an algorithm for identification in text. In *Cambridge Journals: Natural Language Engineering, 3* (Seiten 259 - 289).

Kanejiya, D., Kumar, A., & Prasad, S. (2003). Automatic Evaluation of Students´ Answers using Syntactically Enhanced LSA. In *Proceedings of the HLT-NAACL 2003 workshop on Building educational applications using natural language processing – Volume 2* (Seiten 53 - 60). Morristown: Association for Computational Linguistics.

Karatas, R. (2005). Morphologie: Die Lehre von Bausteinen der Wörter. In Volmert, J. (Hrsg.), *Grundkurs Sprachwissenschaft* (5. Auflage, Seiten 87-98). Stuttgart: Wilhelm Fink Verlag.

Kilgarriff, A. (2010). BNC database and word frequency list. URL - http://www.kilgarriff.co.uk/bnc-readme.html (Zugriffsdatum 29.01.2010)

Klatt, S., & Bohnet, B. (2005). You don´t Have to Think Twice if You Carefully Tokenize. In Su, K. J., Tsujii, J., Lee, J.-H., & Kwong, O. Y. (Hrsg.), In *Natural Language Processing – IJCNLP 2004* (Seiten 299 – 309). Berlin, Heidelberg: Springer Verlag.

Kowalski, G. (1997). *Information Retrieval Systems: Theory and Implementation*. Boston: Kluwer Acadmic Publishers.

Kupiec, J., Petersen, J., & Chen, F. (1999). A trainable document summarizer. In Mani, I., & Maybury, M. (Hrsg.), *Advances in Automatic Text Summarization* (Seiten 55 – 60). Cambridge: MIT Press.

Labadié, A., & Prince, V. (2008). Lexical and Semantic Methods in Inner Text Topic Segmentation: A Comparison between C99 and Transeg. In Kapetanios, E., Sugumaran, V., & Spiliopoulou, M., *Natural Language and Information Systems: 13th International Conference on Applications of Natural Language Processing to Information Systems* (Seiten 347 – 349). Berlin, Heidelberg: Springer Verlag.

Ledeneva, Y., Gelbukh, A., & Garciá-Hernández, R. A. (2008). Terms Derived from Frequent Sequences for Extractive Text Summarization. In Gelbukh, A. (Hrsg.), *Computational Linguistics and Intelligent Text Processing, 9th International Con-*

ference, CICLing 2008, Haifa, Israel, February 2008, Proceedings (Seiten 593 - 604). Berlin, Heidelberg: Springer Verlag.

Li, W., Wong, K.-F., Yuan, C., Li, W., & Xia, Y. (2005). Improving Text Similarity Measurement by Critical Sentence Vector Model. In Lee, G. G., Yamada, A., Meng, H., & Myaeng, S. H. (Hrsg.), *Information Retrieval Technology, Second Asia Information Retrieval Symposium, AIRS 2005, Jeju Island, Korea, October 2005, Proceedings* (Seiten 522 - 527). Berlin, Heidelberg: Springer Verlag.

Lyons, J. (1995). *Theoretical Linguistics* (12. Auflage). Cambridge: Cambridge University Press.

Maas, U. (2006). Phonologie: *Einführung in die funktionale Phonetik des deutschen*. Göttingen: Vandenhoeck & Ruprecht.

Manning, C. D., & Schütze, H. (2003). *Foundations of Statistical Natural Language Processing* (6. überarbeitete Auflage). Cambridge: MIT Press.

Marcu, D. (1999). The automatic construction of large-scale corpora for summarization research. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (Seiten 137 - 144). New York: ACM Press.

Mason, O., & Grove-Stephenson, I. (2002). Automated free text marking with paperless school. In Danson, M. (Hrsg.), *Proceedings of the 6th International Computer Assisted Assessment Conference* (o.S.). Loughborough: Loughborough University.

McCallum, A., & Wellner, B. (2003). *Toward Conditional Models of Identify Uncertainty with Application to Proper Noun Coreference*. URL - http://www.cs.umass.edu/~mccallum/papers/condid-ijcaiws2003.pdf (Zugriffsdatum: 17. Jänner 2010)

McKenna, M. C., Dougherty Stahl, K. A: (2009). *Assessment for Reading Instruction* (2. Auflage). New York: The Guilford Press.

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

Mehler, A. (2005). Gebiete und Phänomene: Text / Fields and phenomena: text. In Köhler, R., Altmann, G., & Piotrowski, R. G. (Hrsg.), *Quantitative Linguistik / Quantitative Linguistics. Ein internationales Handbuch / An international handbook* (Seiten 325 – 347). Berlin: Walter de Gruyter.

Miao, Y (2009). IMS QTI Authoring. In Koper, R. (Hrsg.), *Learning Network Services for Professional Development* (Seiten 389 - 398). Berlin, Heidelberg: Springer Verlag.

Miller, G. A. (1998a). Nouns in WordNet. In Fellbaum, C. (Hrsg.), *WordNet: an electronic lexical database* (Seiten 23 - 45). Massachusetts: MIT Press.

Miller, K. J. (1998b). Modifiers in WordNet. In Fellbaum, C. (Hrsg.), *WordNet: an electronic lexical database* (Seiten 47 - 67). Massachusetts: MIT Press.

Mine, T., Suganuma, A., & Shoudai, T. (2000). The Design and Implementation of Automatic Exercise Generator with Tagged Documents based on the Intelligence of Students: AEGIS. In *Proceedings of the ICCE/ICCAI 2000, 8th International Conference on Computers in Education/International Conference on Computer-Assisted Instruction 2000* (Seiten 651 - 658).

Mine, T., Suganuma, A., & Shoudai, T. (2002). Automatic Generating Appropriate Exercises Based on Dynamic Evaluating both Students' and Questions' Levels. In Parker, P., & Rebelsky, S. (Hrsg.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2002* (Seiten 1898 - 1903). Chesapeake: Association for the Advancement of Computing in Education.

MIT OCW (2010). Chris Hendrikson: Project Management for Construction. URL - http://pmbook.ce.cmu.edu/index.html (Zugriffsdatum 26.03.2010)

Mitkov, R., & Ha, A. L. (2003). Computer-Aided Generation of Multiple-Choice Tests. In *Proceedings of the HLT-NAACL 2003 workshop on Building educational applications using natural language processing* (Seiten 17 - 22). Morristown: Association for Computational Linguistics.

Moens, M.-F., Angheluta, R., & De Busser, R. (2003). Summarization of Texts Found on the World Wide Web. In Abramovicz, W. (Hrsg.), *Knowledge-based Information Retrieval and Filtering from the Web* (Seiten 101 - 120). Dordrecht: Kluwer Academic Publisher Group.

MorphAdorner (2008). Information Technology: MorphAdorner. URL - http://morphadorner.northwestern.edu/ (Zugriffsdatum 20.03.2010)

Multibook (1999). *Multibook: Individuell generierte Lektionen.* URL - http://www.multibook.de (Zugriffsdatum 26.01.2010)

Munoz, R., Saiz-Noeda, M., & Montoyo, A. (2002). Semantic Information in Anaphora Resolution. In Ranchhod, E., & Mamede, N. (Hrsg.), In *Advances in Natural Language Processing: Third International Conference, PorTAL 2002* (Seiten: 63 – 70). Heidelberg: Springer Verlag.

Ouyang, Y., Li, S., & Li, W. (2007). Developing Learning Strategies for Topic-based Summarization. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management* (Seiten 79 - 86). New York: ACM Press.

Pospiech, U. (2005). Syntax: Strukturen in Sätzen. In Volmert, J. (Hrsg.), *Grundkurs Sprachwissenschaft* (5. Auflage, Seiten 115-150). Stuttgart: Wilhelm Fink Verlag.

QTI (2009). *IMS Global Learning Consortium: IMS Question & Test Interoperability Specification, 2010*. URL - http://www.imsglobal.org/question (Zugriffsdatum: 19.01 2010)

QTI (2008). *JQTI*. URL - jqti.qtitools.org (Zugriffsdatum 02.02.2010)

Riloff, E., Wiebe, J., & Wilson, T. (2003). Learning Subjective Nouns using Extraction Pattern Bootstrapping. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 – Volume 4* (Seiten 25 - 32). Morristown: Association for Computational Linguistics.

Roberts, T. S. (2006). Self, Peer and Group Assessment in E-Learning: An Introduction. In Roberts, T. S. (Hrsg.), *Self, Peer and Group Assessment in E-Learning* (Seiten 1 – 16). London: Information Science Publishing.

Rus, V., Cai, Z., & Graesser, A. C. (2007). Experiments on Generating Questions About Facts. In Gelbukh, A. (Hrsg.): *Computational Linguistics and Intelligent Text Processing, 8th International Conference, CICLing 2007, Mexiko City, Mexiko, February 18-24, 2007, Proceedings* (Seiten 444 – 455). Berlin, Heidelberg: Springer Verlag.

Saggion, H. (2005). Topic-based Summarizer at DUC 2005. In *Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing: HLT/EMNLP Document Understanding Conference (DUC) 2005* (o.S.). Morristown: Association for Computational Linguistics.

Schiehlen, M., & Klabunde, R. (2001). Semantik. 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.

Schutz, A. (2008). *SMiLE: XtraK4Me - Extraction of Keyphrases for Metadata Creation*. URL - http://smile.deri.ie/projects/keyphrase-extraction (Zugriffsdatum 15.01.2010)

Song, Y.-I., Han, K.-S., & Rim, H.-C. (2004). A Term Weighting Method based on Lexical Chain for Automatic Summarization. In Gelbukh, A. (Hrsg.): *Computational Linguistics and Intelligent Text Processing, 5th International Conference, CICLing 2004, Seoul, Korea, February 2004, Proceedings* (Seiten 636 – 639). Berlin, Heidelberg: Springer Verlag.

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

Synthetica (2010). Jyloo Software: Synthetica Themes. URL - http://www.jyloo.com/synthetica/themes/ (Zugriffsdatum 03.02.2010)

TAC (2010). *Text Analysis Conference 2009*. URL - http://www.nist.gov/tac/ (Zugriffsdatum 28.01.2009)

Tapanainen, P., & Järvinen, T. (1997). A non-projective dependency parser. In *Proceedings of the 5th Conference of Applied Natural Language Processing (ANLP-5)* (Seiten 64-71). San Francisco: Morgan Kaufmann Publishers.

Tengi, R. I. (1998). Design and implementation of the WordNet Lexical Database and Searching Software. In Fellbaum, C. (Hrsg.), *WordNet: an electronic lexical database* (Seiten 106 - 128). Massachusetts: MIT Press.

TREC (2010). *Text Retrieval Conference (TREC)*. URL - http://trec.nist.gov/ (Zugriffsdatum 11.02.2010)

Valenti, S., Neri, F., & Cucchiarelli, A. (2003). *An Overview of Current Research on Automated Essay Grading. In Journal of Information Technology Education, 2* (Seiten 319 - 330).

Vater, H. (2002). *Einführung in die Sprachwissenschaft*. Paderborn: Wilhelm Fink Verlag.

Vinot, R., & Yvon, F. (2003). Improving Rocchio with Weakly Supervised Clustering. In Lavrac, N., Gamberger, D., Blockeel, H., & Todorovski, L. (Hrsg.), *Machine Learning: ECML 2003, 14th European Conference on Machine Learning, Cavtat-*

Dubrovnik, Croatia, September 2003, Proceedings (Seiten 456 - 467). Berlin, Heidelberg: Springer Verlag.

Weinhofer, J. (2010). *Extraktion semantisch relevanter Daten aus natürlichsprachlichen Inhalten in Hinblick auf eine automatische Fragengenerierung*. (Masterarbeit, TU Graz, IICM, 2010).

Wilcock, G. (2009). *Introduction to Linguistic Annotation and Text Analytics*. Princeton: Morgan & Claypool Publishers.

Williams, D., & Poulovassilis, A. (2008). Combining Data Integration and IE Techniques to Support Partially Structured Data. In Kapetanios, E., Sugumaran, V., & Spiliopoulou, M., *Natural Language and Information Systems: 13th International Conference on Applications of Natural Language Processing to Information Systems* (Seiten 175 – 186). Berlin, Heidelberg: Springer Verlag.

Wiggins, G. P., & McTighe, J. (2005). *Understanding by Design* (2. erweiterte Auflage). Danvers: Association for Supervision and Curriculum Development.

WordNet (2010). About WordNet. URL - http://wordnet.princeton.edu/ (Zugriffsdatum 08.02.2010

10 *Anhang*

10.1 Evaluierung

10.1.1 Evaluierungstext

Organizing for 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 mo-

re 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.

10.1.2 Evaluierungsfragen

Fill In The Blank 1-1:
Subsequently, the functions of project management for construction generally include the following: Specification of _____ 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.

Fill In The Blank 1-2:
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 _____ 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.

Fill In The Blank 1-3:
The management of construction projects requires knowledge of _____ 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.

Fill In The Blank 1-4:
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 _____ 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 _____ may be adapted for project management.

Fill In The Blank 2-1:

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 _____. 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.

Fill In The Blank 2-2:

Hence, management functions can be organized into a hierarchical structure designed to improve operational efficiency. _____ 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.

Fill In The Blank 2-3:

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 _____ 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.

Fill In The Blank 2-4:

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 _____ in current practice, and provide a useful groundwork for project management.

Fill In The Blank 3-1:

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 _____; however, profits derived from earlier facility operation often justify the increase in _____. Generally, if the owner can derive reasonable profits from the operation of a completed facility, the project is considered a success even if _____ far exceed the estimate based on an inadequate scope definition.

Fill In The Blank 3-2:
The programming of capital projects is shaped by the strategic plan of an organization, which is influenced by _____. 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.

Fill In The Blank 3-3:
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 _____ 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.

Fill In The Blank 3-4:
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 _____ however, profits derived from earlier facility operation often justify the increase in _____. Generally, if the owner can derive reasonable profits from the operation of a completed facility, the project is considered a success even if _____ far exceed the estimate based on an inadequate scope definition.

Multiple Choice 1-1:
Subsequently, the functions of project management for construction generally include the following: Specification of _____ 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.
(logging objectives and plans, impression objectives and plans, project objectives and plans, mechanical drawing objectives and plans)

Multiple Choice 1-2:
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 _____ 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.
(modern proficiency techniques, modern management techniques, modern executive techniques, modern teaching method techniques)

Multiple Choice 1-3:
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, _____ and participation satisfaction.

(quantity, human, quality, money)

Multiple Choice 1-4:
Project communications management to ensure effective internal and external communications. Project _____ to analyze and mitigate potential risks. Project procurement management to obtain necessary resources from external sources.
(risk management, time management, cost management, scope management)

Multiple Choice 2-1:
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 _____. 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.
(monte carlo simulation, manager simulation, michael porter simulation, model simulation)

Multiple Choice 2-2:
Hence, management functions can be organized into a hierarchical structure designed to improve operational efficiency. _____ 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 basic documentation functions, The basic serviceablenes functions, The basic chores functions, The basic management functions)

Multiple Choice 2-3:
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 _____ designed to improve operational efficiency. The basic management functions are performed by all managers, regardless of enterprise, activity or hierarchical levels.
(heterarchy structure, exponential stucture, linear structure, hierarchical structure)

Multiple Choice 2-4:
The _____ 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.
(modern management, management process, basic management, general management)

Multiple Choice 3-1:
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 _____; however, profits derived from earlier facility operation often justify the increase in _____. Generally, if the owner can derive reasonable profits from the operation of a completed facility, the project is considered a success even if _____ far exceed the estimate based on an inadequate scope definition.
(gross national products costs, construction costs, expense costs, par value costs)

Multiple Choice 3-2:
The programming of capital projects is shaped by the strategic plan of an organization, which is influenced by _____. 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.
(trade demands and resources constraints, market demands and resources constraints, amount of money demands and resources constraints, labor demands and resources constraints)

Multiple Choice 3-3:
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 _____ 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.
(the investor, market demands, the financial market, market pressure)

Multiple Choice 3-4:
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 _____ 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.
(students, financiers, managers, professionals)

Single Choice 1-1:
Specification of project objectives and plans including delineation of scope, budgeting, scheduling, setting performance inessentials, and selecting project participants.

Single Choice 1-2:
The management of misconstruction projects requires knowledge of modern management as well as an understanding of the design and misconstruction process.

Single Choice 1-3:
There are no similarities between general management of business and industrial corporations and the management of construction projects.

Single Choice 1-4:
Disciplines such as computer science and decision science play no role for project management.

Single Choice 2-1:
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 Michael porter simulation.

Single Choice 2-2:
The basic management functions are performed by all managers, regardless of enterprise, inactiveness or hierarchical levels.

Single Choice 2-3:
The basic management functions are performed by all managers, reagrdless of enterprise, activity or hierarchical levels.

Single Choice 2-4:
A topic of major interest in management science is the minimiation of profit.

Single Choice 3-1:
Invariably, subsequent changes in project scope will increase misconstruction costs ; however, profits derived from earlier facility operation often justify the increase in misconstruction costs.

Single Choice 3-2:
The programming of capital projects is shaped by the strategic plan of an organization, which is influenced by market suppies and resources constraints.

Single Choice 3-3:
With intensive competition for national and international markets, the trend of industrial construction moves toward longer project life cycles, particularly in technology intensive industries.

Single Choice 3-4:
Generally, if the owner can derive reasonable profits from the operation of a completed facility, the project is considered a failure if construction costs far exceed the estimate based on an inadequate scope definition.

Open Ended 1-1:
What do you know about *project objectives and plans* in the context of *Project Management*?

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.

Open Ended 1-2:

What do you know about *modern management techniques* in the context of *Project Management*?

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.

Open Ended 1-3:

What is *Project Management*?

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.

Open Ended 1-4:

What are *the four functions* of *project management for construction*?

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.

Open Ended 2-1:

What do you know about *Monte Carlo simulation* subjected to *Trends in Modern Management*?

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.

Open Ended 2-2:

What do you know about *management functions* in the context of *Trends in Modern Management*?

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.

Open Ended 2-3:

Which *operation research techniques* are *used for optimization*?

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.

Open Ended 2-4:

What does strategic positioning require?

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

Open Ended 3-1:

What do you know about *construction costs* in the context of *Strategic Planning and Project Programming*?

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.

Open Ended 3-2:

What do you know about *market demands and resources constraints* in the context of *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.

Open Ended 3-3:

What is the *trend of industrial construction*?

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.

Open Ended 3-4:

What is *the role of the owner or facility sponsor* in construction process?

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.

10.1.3 Evaluierungsergebnisse

Pertinence computerized	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Fill In The Blank 1-1	100	80	90	100	100	94	8,94
Fill In The Blank 1-2	100	80	80	50	100	82	20,49
Fill In The Blank 2-1	100	70	50	100	100	84	23,02
Fill In The Blank 2-2	90	70	100	75	85	84	11,94
Fill In The Blank 3-1	100	80	90	100	100	94	8,94
Fill In The Blank 3-2	100	80	40	75	100	79	24,60
	98,33	**76,67**	**75,00**	**83,33**	**97,50**	**86,17**	

Level computerized	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Fill In The Blank 1-1	75	50	60	100	50	67	21,10
Fill In The Blank 1-2	75	40	80	60	50	61	16,73
Fill In The Blank 2-1	75	50	70	100	50	69	20,74
Fill In The Blank 2-2	75	50	80	75	50	66	14,75
Fill In The Blank 3-1	90	50	60	50	50	60	17,32
Fill In The Blank 3-2	95	50	80	100	50	75	23,98
	80,83	**48,33**	**71,67**	**80,83**	**50,00**	**66,33**	

Concept computerized	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Fill In The Blank 1-1	100	80	90	25	100	79	31,30
Fill In The Blank 1-2	100	20	90	10	100	64	45,06
Fill In The Blank 2-1	85	70	80	90	100	85	11,18
Fill In The Blank 2-2	95	50	100	85	80	82	19,56
Fill In The Blank 3-1	90	90	90	80	100	90	7,07
Fill In The Blank 3-2	100	90	50	75	100	83	21,10
	95,00	**66,67**	**83,33**	**60,83**	**96,67**	**80,50**	

Pertinence manual	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Fill In The Blank 1-3	85	70	90	30	100	75	27,39
Fill In The Blank 1-4	100	60	100	50	75	77	22,80
Fill In The Blank 2-3	90	60	40	50	70	62	19,24
Fill In The Blank 2-4	100	80	20	100	50	70	34,64
Fill In The Blank 3-3	100	80	90	100	100	94	8,94
Fill In The Blank 3-4	100	80	90	100	100	94	8,94
	95,83	**71,67**	**71,67**	**71,67**	**82,50**	**78,67**	

Level manual	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Fill In The Blank 1-3	80	50	50	30	50	52	17,89
Fill In The Blank 1-4	60	50	70	35	50	53	13,04
Fill In The Blank 2-3	100	50	70	75	50	69	20,74
Fill In The Blank 2-4	90	50	80	50	50	64	19,49
Fill In The Blank 3-3	90	50	50	15	50	51	26,55
Fill In The Blank 3-4	90	50	60	50	50	60	17,32
	85,00	**50,00**	**63,33**	**42,50**	**50,00**	**58,17**	

Concept manual	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Fill In The Blank 1-3	95	20	90	10	100	63	44,10
Fill In The Blank 1-4	100	20	90	10	75	59	41,29
Fill In The Blank 2-3	100	60	50	75	70	71	18,84
Fill In The Blank 2-4	100	80	30	75	50	67	27,29
Fill In The Blank 3-3	95	90	90	75	100	90	9,35
Fill In The Blank 3-4	90	90	90	80	100	90	7,07
	96,67	**60,00**	**73,33**	**54,17**	**82,50**	**73,33**	

Pertinence computerized	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Multiple Choice 1-1	100	70	50	85	100	81	21,33
Multiple Choice 1-2	100	70	80	75	100	85	14,14
Multiple Choice 2-1	100	90	20	100	80	78	33,47
Multiple Choice 2-2	90	60	90	75	100	83	15,65
Multiple Choice 3-1	100	80	90	100	100	94	8,94
Multiple Choice 3-2	100	80	40	75	100	79	24,60
	98,33	**75,00**	**61,67**	**85,00**	**96,67**	**83,33**	

Level computerized	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Multiple Choice 1-1	75	60	50	75	50	62	12,55
Multiple Choice 1-2	75	50	90	70	50	67	17,18
Multiple Choice 2-1	30	50	80	100	50	62	27,75
Multiple Choice 2-2	75	50	50	75	40	58	16,05
Multiple Choice 3-1	90	50	80	50	50	64	19,49
Multiple Choice 3-2	90	50	50	100	50	68	24,90
	72,50	**51,67**	**66,67**	**78,33**	**48,33**	**63,50**	

Distractors computerized	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Multiple Choice 1-1	100	70	90	100	40	80	25,50
Multiple Choice 1-2	100	70	100	90	90	90	12,25
Multiple Choice 2-1	50	80	30	50	80	58	21,68
Multiple Choice 2-2	75	70	100	50	80	75	18,03
Multiple Choice 3-1	95	90	100	60	50	79	22,47
Multiple Choice 3-2	90	90	90	75	80	85	7,07
	85,00	**78,33**	**85,00**	**70,83**	**70,00**	**77,83**	

Pertinence manual	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Multiple Choice 1-3	100	90	40	100	60	78	26,83
Multiple Choice 1-4	95	70	80	100	100	89	13,42
Multiple Choice 2-3	100	60	100	100	100	92	17,89
Multiple Choice 2-4	95	40	100	100	100	87	26,36
Multiple Choice 3-3	100	80	90	100	100	94	8,94
Multiple Choice 3-4	100	70	40	30	100	68	32,71
	98,33	**68,33**	**75,00**	**88,33**	**93,33**	**84,67**	

Level manual	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Multiple Choice 1-3	90	50	80	60	50	66	18,17
Multiple Choice 1-4	60	70	40	60	50	56	11,40

Multiple Choice 2-3	85	50	70	50	50	61	15,97
Multiple Choice 2-4	85	50	70	80	50	67	16,43
Multiple Choice 3-3	90	50	80	15	50	57	29,50
Multiple Choice 3-4	95	50	40	5	50	48	32,13
	84,17	**53,33**	**63,33**	**45,00**	**50,00**	**59,17**	

Distractors manual	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Multiple Choice 1-3	85	80	50	40	70	65	19,36
Multiple Choice 1-4	100	80	40	100	90	82	24,90
Multiple Choice 2-3	100	70	100	45	80	79	23,02
Multiple Choice 2-4	100	30	100	100	100	86	31,30
Multiple Choice 3-3	90	90	90	25	100	79	30,50
Multiple Choice 3-4	100	80	30	5	100	63	43,24
	95,83	**71,67**	**68,33**	**52,50**	**90,00**	**75,67**	

Pertinence computerized	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Single Choice 1-1	100	80	70	100	70	84	15,17
Single Choice 1-2	90	70	60	75	70	73	10,95
Single Choice 2-1	100	80	70	100	80	86	13,42
Single Choice 2-2	90	60	80	90	80	80	12,25
Single Choice 3-1	95	70	70	50	100	77	20,49
Single Choice 3-2	95	80	80	50	100	81	19,49
	95,00	**73,33**	**71,67**	**77,50**	**83,33**	**80,17**	

Level computerized	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Single Choice 1-1	100	50	90	100	20	72	35,64
Single Choice 1-2	50	50	70	25	20	43	20,49
Single Choice 2-1	75	40	70	90	50	65	20,00
Single Choice 2-2	95	20	70	25	50	52	31,34
Single Choice 3-1	90	30	40	25	40	45	25,98
Single Choice 3-2	70	50	50	25	30	45	18,03
	80,00	**40,00**	**65,00**	**48,33**	**35,00**	**53,67**	

Pertinence manual	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Single Choice 1-3	90	70	90	75	30	71	24,60
Single Choice 1-4	90	70	80	50	50	68	17,89
Single Choice 2-3	90	60	80	90	80	80	12,25
Single Choice 2-4	65	50	60	25	80	56	20,43
Single Choice 3-3	95	100	80	100	100	95	8,66
Single Choice 3-4	100	80	60	100	70	82	17,89
	88,33	**71,67**	**75,00**	**73,33**	**68,33**	**75,33**	

Level manual	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Single Choice 1-3	65	50	90	90	80	75	17,32
Single Choice 1-4	70	30	60	5	100	53	36,67
Single Choice 2-3	95	50	70	25	50	58	26,12
Single Choice 2-4	25	0	30	0	40	19	18,17
Single Choice 3-3	100	50	70	75	70	73	17,89

Single Choice 3-4	95	50	70	20	40	55	28,72
	75,00	**38,33**	**65,00**	**35,83**	**63,33**	**55,50**	

Pertinence computerized	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Open Ended 1-1	100	80	100	100	80	92	10,95
Open Ended 1-2	100	80	90	100	80	90	10,00
Open Ended 2-1	100	80	100	90	20	78	33,47
Open Ended 2-2	100	80	100	100	100	96	8,94
Open Ended 3-1	100	90	100	100	100	98	4,47
Open Ended 3-2	100	90	90	100	75	91	10,25
	100,00	**83,33**	**96,67**	**98,33**	**75,83**	**90,83**	

Level computerized	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Open Ended 1-1	90	50	50	100	100	78	25,88
Open Ended 1-2	90	50	80	75	100	79	18,84
Open Ended 2-1	80	50	70	100	50	70	21,21
Open Ended 2-2	80	50	40	90	50	62	21,68
Open Ended 3-1	90	50	50	75	50	63	18,57
Open Ended 3-2	95	50	60	100	50	71	24,60
	87,50	**50,00**	**58,33**	**90,00**	**66,67**	**70,50**	

Answer computerized	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Open Ended 1-1	95	90	70	100	80	87	12,04
Open Ended 1-2	80	90	70	100	80	84	11,40
Open Ended 2-1	95	100	100	100	20	83	35,28
Open Ended 2-2	95	80	80	95	100	90	9,35
Open Ended 3-1	95	90	90	100	100	95	5,00
Open Ended 3-2	100	80	80	100	100	92	10,95
	93,33	**88,33**	**81,67**	**99,17**	**80,00**	**88,50**	

Pertinence manual	User 1	User 2	User 3	User 4	User 5	Average	Std. Dev.
Open Ended 1-3	100	80	100	100	100	96	8,94
Open Ended 1-4	100	70	90	75	100	87	13,96
Open Ended 2-3	100	70	100	100	100	94	13,42
Open Ended 2-4	100	100	60	75	100	87	18,57
Open Ended 3-3	100	100	100	100	100	100	0,00
Open Ended 3-4	90	70	80	85	100	85	11,18
	98,33	**81,67**	**88,33**	**89,17**	**100,00**	**91,50**	

10.2 CD