Mag. Matthias Reichhold

ROBUS: Personalisierte Suche in natürlichsprachlichen Unternehmensdaten

DISSERTATION

zur Erlangung des akademischen Grades
Doktor der Technischen Wissenschaften

Alpen-Adria-Universität Klagenfurt
Fakultät für Technische Wissenschaften

GUTACHTER

O. Univ.-Prof. Dipl.-Ing. Dr. Dr. h.c. Heinrich C. Mayr
Institut für Angewandte Informatik

Ao. Univ.-Prof. Dr. Günther Fliedl
Institut für Angewandte Informatik

Klagenfurt, Juni 2014

Ehrenwörtliche Erklärung

Ich erkläre ehrenwörtlich, dass ich die vorliegende wissenschaftliche Arbeit selbstständig angefertigt und die mit ihr unmittelbar verbundenen Tätigkeiten selbst erbracht habe. Ich erkläre weiters, dass ich keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle ausgedruckten, ungedruckten oder dem Internet im Wortlaut oder im wesentlichen Inhalt übernommenen Formulierungen und Konzepte sind gemäß den Regeln für wissenschaftliche Arbeiten zitiert und durch Fußnoten bzw. durch andere genaue Quellenangaben gekennzeichnet.

Die während des Arbeitsvorganges gewährte Unterstützung einschließlich signifikanter Betreuungshinweise ist vollständig angegeben.

Die wissenschaftliche Arbeit ist noch keiner anderen Prüfungsbehörde vorgelegt worden.

Diese Arbeit wurde in gedruckter und elektronischer Form abgegeben. Ich bestätige, dass der Inhalt der digitalen Version vollständig mit dem der gedruckten Version übereinstimmt.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Matthias Reichhold, Klagenfurt, Juni 2014

Danksagung

Mein besonderer Dank gilt Herrn O. Univ.-Prof. Dipl.-Ing. Dr. Dr. hc. Heinrich C. Mayr und Herrn Ao. Univ.-Prof. Dr. Günther Fliedl für die fachliche Betreuung meiner Arbeit. Ohne ihre Unterstützung und wertvollen Ratschläge hätte ich die Arbeit nicht in dieser Form erstellen können. Herrn Dr. Christian Winkler und Herrn Dr. Jörg Kerschbaumer danke ich für die ausgezeichnete Zusammenarbeit im Zuge unserer Forschungstätigkeiten und Publikationen. Für die programmiertechnische Unterstützung bin ich Herrn Clemens Eberwein und Herrn Michael Brodskiy zu Dank verpflichtet. Bei Dr.in Petra Ziegler von der 3s Unternehmensberatung bedanke ich mich für die Bereitstellung des DISCO Thesaurus. Weiters möchte ich Herrn Kurt Siegl und Herrn Thomas Kalcher für ihre Beiträge aus der Unternehmensperspektive danken – diese waren speziell bei der Analyse der Ausgangsituation und der Zielsetzung von zentraler Bedeutung.

Mein Dank gebührt auch meiner Familie, allen voran meinen Eltern, Margit und Mathias Reichhold, die mich stets in meinen Bestrebungen nach besten Kräften unterstützt haben. Abschließend möchte ich meiner Lebensgefährtin, Christina Schürz, ganz besonders danken. Sie hat nicht nur wesentlich dazu beigetragen, diese Arbeit von stilistischen, grammatikalischen und rechtschreibtechnischen Irrwegen fern zu halten, sondern hat mich über all die Jahre in meiner Arbeit bestärkt und war so eine unerlässliche Stütze, ohne die ich diese Arbeit nie zu Ende führen hätte können.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Gleichungsverzeichnis

Kurzfassung

Diese Arbeit stellt ein neuartiges Verfahren zur Optimierung von Suchaufgaben in unstrukturierten Unternehmensdaten auf Basis von automatisch generierten Rollenprofilen vor. Es wird gezeigt, wie Rollenprofile mit Hilfe eines standardisierten Thesaurus (Müller-Riedlhuber 2009) aus internetbasierten Stellenausschreibungen extrahiert werden können.

Hierfür kommen spezielle Methoden der Computerlinguistik zum Einsatz, die es ermöglichen, Rollenprofile in Form von gewichteten Termvektoren textuell zu beschreiben. In einem ersten Schritt wird ein Textkorpus mit mehreren Tausend englischsprachigen Stellenausschreibungen erstellt. Die dafür verwendeten Rohdaten werden von der Internetplattform LinkedIn[1] bezogen und können von dort mit speziellen Web-Services ausgelesen werden. Nach einem initialen Bereinigungsschritt (entfernen von Formatierungszeichen, etc.) werden die Texte in einzelne Segmente („Tokens“) aufgeteilt. Für jedes Token wird anschließend mit Hilfe eines Part-Of-Speech-Taggers (Apache-OpenNLP-Development-Community 2013b) die entsprechende Wortart bestimmt und ein Abgleich mit dem bereits erwähnten Thesaurus durchgeführt. Auf Basis dieses Abgleichs und weiterer Kriterien (Position und Häufigkeit des Tokens innerhalb des Ausschreibungstextes) wird der Term entweder als nicht relevant verworfen oder dem Rollenvektor hinzugefügt. Darüber hinaus wird ein Mechanismus zur dynamischen Zuordnung von generierten Rollenprofilen zu textuellen Daten vorgestellt. Dabei ist zu beachten, dass der gewählte Mechanismus (1) gänzlich ohne manuelles Zutun funktioniert und (2) sich über die dynamische Zuordnung bestimmen lässt, wie relevant ein textuelles Dokument für eine bestimmte Rolle ist. Dies erfolgt durch die Bestimmung von Ähnlichkeiten im Vektorraum-

1 https://developer.linkedin.com/apis#jobs

Modell. Dabei wird für jeden Rollenvektor die Kosinus-Ähnlichkeit zu den vorhandenen Textdokumenten berechnet (Chim & Deng 2007; Manning & Schuetze 1999).
Die Optimierung der Suchergebnisse erfolgt auf Basis folgender These: Umso höher die Ähnlichkeit eines Dokuments zu einem Rollenprofil ist, desto höher ist auch die Relevanz des jeweiligen Dokuments für alle Benutzer/innen, die diesem Rollenprofil zugeordnet sind. Dementsprechend werden die Suchergebnisse anhand der Ähnlichkeitswerte neu gereiht.
Im Rahmen der Dissertation wurde das hier beschriebene Verfahren vollständig implementiert und ausführlich evaluiert. Auf Basis der daraus gewonnen Testergebnisse wurde bewiesen, dass ein rollensensitives Suchverfahren in Form des ROBUS Systems zu einer signifikanten Verbesserung im Bereich der Unternehmenssuche beiträgt. Zu beachten ist außerdem, dass dieses System nicht als vollständige Suchmaschine, sondern vielmehr als Erweiterung zu bestehenden Suchsystemen zu verstehen ist. Die Suchergebnisse der zugrunde liegenden Suchmaschine werden durch das System anhand der Rollenprofile neu gereiht. Zur Evaluierung des Systems wurde die frei verfügbare Apache Lucene Implementierung herangezogen. Diese Vorgehensweise ermöglicht die Generierung von objektiven und vergleichbaren Testergebnissen.

Abstract

This thesis introduces a novel method for the optimization of search tasks for unstructured data in enterprise information systems based on automatically generated role profiles. It is shown how role profiles can be extracted using a standardized thesaurus (Müller-Riedlhuber 2009) from online job postings. For this purpose special methods from the area of computational linguistics are applied, which provide the means to textually describe role profiles in the form of weighted term vectors. In a first step, a text corpus with several thousand English job postings is created. For this purpose, raw data is obtained via special web services from the Internet platform LinkedIn. After the first step of data cleansing (i.e., removing formatting characters, etc.) the text stream is divided into individual segments ("tokens"). For each token the corresponding part of speech is determined using a part-of-speech tagger (Apache-OpenNLP-Development-Community 2013b) and a lookup operation with the above mentioned thesaurus performed. Based on the lookup operation and other criteria (position and frequency of the token within the text) a term is either discarded as irrelevant or added to the role vector. In addition, a mechanism for a dynamic assignment of generated role profiles to textual data is presented. It should be noted that (1) the chosen mechanism works completely without manual intervention and that (2) the relevance of a textual document for a particular role can be determined on the basis of this assignment. This is accomplished by the computation of similarities in the vector space model. The cosine similarity is calculated for every text document and for each role vector (Chim & Deng 2007; Manning & Schuetze 1999).

The optimization of the search results is based on the following argument: The greater the similarity between a document and a role profile, the higher the relevance of that document for all users who are assigned to the role profile. Accordingly, the search results are re-ranked based on the similarity values.

The method described here has been fully implemented and evaluated in detail as part of this dissertation. Based on the obtained test results it has been proven that a role-sensitive search method in the form of ROBUS contributes to a significant improvement in the area of enterprise search. It should also be noted that this system is not to be understood as a complete search engine, but rather as an extension to existing search systems. The search results of the underlying search engine are re-ranked by the system using the role profiles. For evaluation purposes the freely available Apache Lucene implementation was used as a baseline system. This approach allows the generation of objective and comparable test results.

1 Einleitung

1.1 Motivation

Die Menge an Daten, mit denen Wissensarbeiter/innen heutzutage in Unternehmen konfrontiert sind, steigt rapide an. Den größten und zudem am schnellsten wachsenden Teil machen dabei unstrukturierte (textuelle) Daten aus (Dou et al. 2007; Huang et al. 2010). Darin können Mitarbeiter/innen Informationen bzw. Dokumente immer schwerer finden. Aus diesem Grund gewinnen jene unternehmensinternen Suchmaschinen an Bedeutung, die in der Lage sind, unstrukturierte Daten zu verarbeiten (E Agichtein et al. 2006). Im Gegensatz zum Bereich "Internet-Suche" wurde dem Thema "Unternehmenssuche" bisher aber vergleichsweise wenig Beachtung geschenkt (E Agichtein et al. 2006).
In der Forschung ist man sich darüber einig, dass die Verwendung von kontextuellen Informationen, zusätzlich zu den eingegebenen Suchbegriffen, maßgeblich zur Verbesserung von Unternehmenssuchmaschinen beitragen kann. Im Unternehmensumfeld können Daten über Mitarbeiter und Mitarbeiterinnen sowie deren berufliche Rollen herangezogen werden, um solche kontextuellen Informationen zu gewinnen (Papacharissi 2009). Suchsysteme für Unternehmensdaten weisen jedoch erhebliche Unterschiede zu Internet-Suchmaschinen auf und müssen drei wesentliche Bereiche abdecken: (1) Durchsuchen der externen (öffentlich zugänglichen) Firmen-Webseite, (2) Durchsuchen der internen Firmen-Webseite ("Intranet") und (3) Durchsuchen von unternehmensinternen Dokumenten, Datenbankeinträgen, E-Mail-Nachrichten und sonstigen textuellen Datensammlungen (Cleverdon 1991).

Ziel dieser Arbeit war es, ein System zu entwickeln, das die individuellen Rollen von Mitarbeiter/innen in einem Unternehmen als kontextuelle Zusatzinformation bei der Suche in unstrukturierten Dokumenten miteinbezieht und so die Informationsbedürfnisse der Anwender/innen besser befriedigen kann. Weiters galt es zu untersuchen, ob bzw. in welchem Maße eine rollensensitive Adaption von Suchergebnissen zu einer Steigerung der Qualität eines Informationssuchsystems im Unternehmensumfeld beitragen kann. Die konkreten Forschungsfragen werden im folgenden Kapitel näher erläutert.

1.2 Forschungsfragen

Im Rahmen dieser Arbeit werden die nachfolgend beschriebenen Forschungsfragen diskutiert und anhand einer konkreten Systemimplementierung („ROBUS“) evaluiert. Sie basieren auf der eingangs vorgestellten Annahme, dass Suchsysteme für unstrukturierte Unternehmensdaten optimiert werden können, indem sie die jeweilige Rolle miteinbeziehen, die der suchende Benutzer im Unternehmen einnimmt.

1.2.1 Zusammenhang zwischen Rollenprofilen und Dokumentinhalten

Die erste und grundlegendste Frage zur Untersuchung der o.a. Annahme lautet:
„Wie kann die berufliche Rolle, die ein/e Mitarbeiter/in in einem Unternehmen inne hat (z.B. „Software Developer“, „Marketing Leiter/in“ oder „Vertriebsinnendienst“), mit den zu durchsuchenden Dokumenten in Relation gebracht werden?“

Zwar sind auf dem Gebiet der personalisierten bzw. kontextsensitiven Suche allgemein rege Forschungsaktivitäten zu beobachten; dem Thema der Unternehmensrollen als Kontextinformation schenkte die Wissenschaft bisher jedoch kaum Aufmerksamkeit. Eine Ausnahme bildet die Arbeit von (Kohn et al. 2008), im Rahmen derer die Reihung von Suchergebnissen im Datenbestand eines Pharma-Unternehmens durch Benutzerrollen beeinflusst wird. Das System arbeitet dabei jedoch nicht mit Unternehmensrollen im Sinne der individuellen Tätigkeiten, vielmehr konstruiert es benutzerspezifische Profile, die anhand verschiedener Eigenschaften der jeweiligen Mitarbeiter/innen konstruiert werden.
Im Gegensatz dazu wird in der vorliegenden Arbeit eine neuartige Methode präsentiert, die den Zusammenhang zwischen Unternehmensrollen und Stellenausschreibungen nutzt, um in den Ausschreibungstexten die die Unternehmensrolle am besten charakterisierenden Begriffe zu identifizieren und daraus personenübergreifende Rollenprofile zu extrahieren. Die Rollenprofile wiederum können im Vektorraum-Modell abgebildet und auf diese Weise mit den textuellen Dokumentinhalten in Relation gesetzt werden (siehe Kapitel 5.1).

1.2.2 Automatische Erstellung von Rollenprofilen

Um organisationsspezifische Rollenprofile als Quelle kontextueller Informationen in einem Unternehmenssuchsystem einbeziehen zu können, bedarf es einer Methode, mit Hilfe derer die benötigten Profilvektoren automatisiert erstellt und den jeweiligen Benutzer/inne/n zugewiesen werden können. Dazu müssen die Quelldaten, die in Form von Stellenausschreibungstexten in natürlicher Sprache vorliegen, eingelesen, bereinigt und weiterverarbeitet werden können.
Des Weiteren müssen aus der Menge aller Wörter in den Ausschreibungstexten jene identifiziert werden, die die jeweilige Rolle am besten charakterisieren, während umgekehrt alle „allgemeinen“, d.h. nicht rollenspezifischen Wörter herausgefiltert werden müssen.
Daher lautet die zweite Forschungsfrage dieser Arbeit:

„Mit welchen konkreten Methoden können die unstrukturierten Stellenausschreibungstexte automatisiert verarbeitet und die relevanten Rollenbegriffe daraus extrahiert werden?“.

Im Rahmen von ROBUS wird dafür eine eigens entwickelte Vorgehensweise präsentiert. Diese erstellt mit Hilfe einer weitreichenden computerlinguistischen Analyse, einem domänenspezifischen, multilingualen Thesaurus sowie einer speziellen Gewichtungsfunktion für jede definierte Unternehmensrolle ein zugehöriges Rollenprofil. Die dafür benötigten grundlegenden Komponenten aus dem Bereich der Computerlinguistik werden in Kapitel 0 erläutert. Die konkrete Anwendung bzw. Vorgehensweise im ROBUS Verfahren wird in Kapitel 5.1.4 präsentiert.

1.2.3 Rollensensitive Reihung von Suchergebnissen

Informationssysteme im Unternehmensumfeld berücksichtigen vielfach keinerlei kontextuelle Informationen, sondern verwenden ausschließlich die vom Benutzer eingegebene Suchanfrage zur Interpretation des aktuellen Informationsbedürfnisses. Die benutzerspezifischen Anforderungen werden dabei nicht berücksichtigt. Vielmehr gilt das „One-Fits-4-All“ Paradigma, im Zuge dessen jeder Anwender für die gleiche Suchanfrage genau das gleiche Suchergebnis vom System erhält. (Demartini 2007)

Dieses Problem wird im Rahmen der folgenden Forschungsfrage untersucht:

„ Wie können die vom ROBUS Verfahren generierten Rollenprofile eingesetzt werden, um die Suchergebnisse hinsichtlich der Unternehmensrolle des suchenden Benutzers/der suchenden Benutzerin anzupassen?“

Die rollensensitive Anpassung der Ergebnisreihung erfolgt dabei abhängig von der Relevanz, die ein bestimmtes Dokument in Bezug auf eine Unternehmensrolle hat, wobei Suchergebnisse mit höheren Relevanzwerten weiter nach oben gereiht werden und vice versa. Durch diese Methodik wird gewährleistet, dass die rollenabhängigen Informationsbedürfnisse des jeweiligen Benutzers bei der Suche einfließen und das „One-Fits-4-All“

Problem behoben wird. Die konkrete Vorgehensweise zur Berechnung der Relevanz eines Dokuments für eine Rolle wird in Kapitel 5.2.1 dargestellt. Die Funktion zur rollensensitiven Reihung von Suchergebnissen beschreibt Kapitel 5.2.2.

1.2.4 Evaluation von rollensensitiven Suchsystemen

Die vierte und abschließende Forschungsfrage lautet:
„Welche Verfahren und Rahmenbedingungen werden für eine objektive und rekonstruierbare Evaluation von kontextsensitiven Systemen im Allgemeinen und von ROBUS im Speziellen benötigt?“
Die Evaluation von Suchsystemen ist allgemein kein triviales Unterfangen. Dies gilt besonders für kontextsensitive Systeme, da hierbei die kontextabhängigen Parameter in die Evaluation miteinfließen müssen. In der Literatur existieren umfangreiche Ressourcen zur Evaluation von nicht-kontextsensitiven Systemen. So liefert z.B. die TREC Reihe ein ganzes Sortiment an Testdatensätzen (Korpora, Suchanfragen und Relevanzbewertungen) für die unterschiedlichen Anwendungsbereiche der Informationssuche (NIST 2010). Neben dem deutlich verminderten Aufwand zur Erlangung von geeigneten Evaluationsdaten birgt die Verwendung solch standardisierter Datensätze auch den Vorteil, dass Evaluationsergebnisse von verschiedenen Suchsystemen miteinander verglichen werden können (vgl. Kapitel 3.3).
Standardisierte Testdaten für kontextsensitive bzw. personalisierte Suchsysteme stehen im Vergleich dazu in einem deutlich eingeschränkteren Umfang zur Verfügung (vgl. Kapitel 3.5). Eine Datensammlung, die zur Evaluation eines rollensensitiven Suchsystems wie ROBUS geeignet wäre, war bis dato nicht bekannt bzw. existent.
Die gegenständliche Arbeit hatte daher auch zum Ziel, die Anforderungen und Lösungsmöglichkeiten in Bezug auf die Evaluation eines rollensensitiven Informationssuchsystems im Unternehmensumfeld zu analysieren und umzusetzen. Die diesbezüglichen Erkenntnisse werden in Kapitel 6.1 präsentiert.

1.3 Überblick

Die vorliegende Arbeit dokumentiert das rollenbasierte Suchverfahren ROBUS, mit Hilfe dessen die Suche nach Informationen in unstrukturierten Daten im Unternehmensumfeld erleichtert werden soll, indem die langfristigen Informationsbedürfnisse der Benutzer/innen in Form ihrer Rolle innerhalb des Unternehmens berücksichtigt werden. Die Arbeit gliedert sich wie folgt:

Kapitel 0 beginnt mit einer kurzen Einführung in das Thema Informationssuche und beschreibt anschließend einige wesentliche Konzepte von modernen Suchsystemen wie beispielsweise das Vektorraum-Modell oder das BM25 Verfahren. Der Fokus liegt dabei vor allem auf jenen Konzepten, die auch im ROBUS Verfahren selbst Anwendung finden.

Kapitel 0 widmet sich der Evaluation von Suchsystemen. Ausgehend von den grundlegenden Prinzipen bei der Planung, Durchführung und Auswertung solcher Systemtests werden die Anforderungen und Möglichkeiten zur Evaluation von kontextsensitiven und personalisierten Suchsystemen beschrieben. Ein Schwerpunkt des Kapitels liegt auf der Verwendung von sogenannten „Folksonomies“ zur Gewinnung von personalisierten Relevanzbewertungen.

Die Analyse von natürlich-sprachlichen Texten mit Hilfe von computerlinguistischen Methoden wird in Kapitel 0 erläutert. Systemrelevante Funktionen wie z.B. die Tokenisierung werden darin beschrieben. Des Weiteren enthält das Kapitel Informationen zu linguistischen Ressourcensammlungen (z.B. Roget Thesaurus, WordNet, u.a.) und beschreibt einen speziellen Thesaurus („DISCO“), der von ROBUS zur Erstellung der Rollenprofile verwendet wird.

Die konkrete Vorgehens- und Funktionsweise von ROBUS wird in Kapitel 0 beschrieben. Der erste Teil des Kapitels widmet sich dem Thema der automatisierten Erstellung von Rollenprofilen. Dabei wird gezeigt, wie Stellenausschreibungstexte mittels computerlinguistischer Methoden analysiert und zur Gewinnung von rollenspezifischen Begriffen verwendet werden. Außerdem wird die Vorgehensweise zur Repräsentation von Rollenprofilen in Form von gewichteten Termvektoren erläutert.

Der zweite Teil des Kapitels dokumentiert die Methodik zur Berechnung jener Relevanzwerte, die Auskunft darüber geben, wie interessant ein bestimmtes Dokument für eine

bestimmte Unternehmensrolle ist. Des Weiteren wird gezeigt, wie eine Menge an Suchergebnissen anhand dieser Relevanzwerte gereiht werden kann, sodass die rollenbasierten Informationsbedürfnisse der Benutzer/innen berücksichtigt werden können.
Kapitel 0 enthält eine detaillierte Beschreibung der Evaluationsmethode und Testumgebung auf Grundlage derer die Funktionsweise des ROBUS Systems überprüft wurde. Darüber hinaus werden die konkreten Testergebnisse präsentiert.
Kapitel 7 liefert eine abschließende Zusammenfassung der Arbeit sowie einen Ausblick in Bezug auf weiterführende Forschungsaktivitäten.

2 Aktuelle Technologien in der Informationssuche

Die computerunterstützte Suche nach Informationen in unstrukturierten, textuellen Daten hat in den letzten Jahren stark an Bedeutung gewonnen. Die stetig steigende Menge an Dokumenten bedingt spezielle Informationssuchsysteme, häufig auch als „Suchmaschinen“ bezeichnet, die in der Lage sind, die Informationsbedürfnisse der jeweiligen Benutzer/innen zu befriedigen. Dies hat zur Entwicklung einer Vielzahl von unterschiedlichen Systemen und Methoden sowie zu einer Spezialisierung auf dem Gebiet der Informationssuche geführt. So gibt es beispielsweise Suchsysteme, die sich ausschließlich auf die Suche im Internet („Web Search“) konzentrieren und darüber hinaus kontextsensitive Zusatzinformationen („Personalized Search“) miteinfließen lassen (Bouadjenek et al. 2013). Wieder andere Systeme fokussieren sich auf den Bereich der unternehmensinternen Suche („Enterprise Search“), da sich die Informationsbedürfnisse von Mitarbeitern/innen in einem Unternehmen grundlegend von jenen unterscheiden, die Anwender/innen einer Internetsuchmaschine haben (Demartini 2007).

Auch das in dieser Arbeit präsentierte ROBUS System stellt ein hochspezialisiertes Informationssuchsystem dar, das eigens für den Einsatz im Unternehmensbereich konzipiert und dahingehend optimiert wurde. Um den besonderen Anforderungen bei der Suche in unstrukturierten Unternehmensdaten Rechnung zu tragen, generiert ROBUS unternehmensspezifische Rollenprofile, die den jeweiligen Benutzer/inne/n zugewiesen werden. Das ROBUS System ist dadurch bei der Interpretation der Informationsbedürfnisse nicht mehr nur auf die eingegebene Suchanfrage beschränkt, sondern kann zusätzlich das zugewiesene Rollenprofil miteinbeziehen.

Trotz der zahllosen Weiterentwicklungen und Spezialisierungen der letzten Jahre existieren nach wie vor einige grundlegende Technologien, die in nahezu jedem Informationssuchsystem angewandt werden und die die Basis für die darauf aufbauenden Optimierungen darstellen.
In diesem Kapitel werden die wesentlichen Technologien aus dem Bereich der Informationssuche, die auch von ROBUS verwendet werden bzw. für das System von Relevanz sind, beschrieben. Dazu zählen (1) das Vektorraum-Modell als grundlegende Basis zur Repräsentation von Dokumenten und Suchanfragen in Form von gewichteten Termvektoren (Kapitel 2.1), (2) das *tf-idf* Maß zur Gewichtung bzw. quantitativen Bewertung der Bedeutung eines Terms für ein Dokument (Kapitel 2.2) und (3) die Kosinus-Ähnlichkeit zur Beurteilung der Übereinstimmung (Ähnlichkeit) von Dokumenten untereinander bzw. von Suchanfragen und Dokumenten (Kapitel 2.3). Des Weiteren wird die BM25 Methode als Vertreter der probabilistischen Abfragemodelle vorgestellt und ihr Einsatz im Rahmen des ROBUS Projekts erläutert (Kapitel 2.4).

2.1 Das Vektorraum-Modell

Das Vektorraum-Modell, besser bekannt unter der englischen Bezeichnung VECTOR SPACE MODEL, wird im Bereich der Informationssuche zur Abbildung von textuellen Dokumenten und Suchanfragen in Form von gewichteten Vektoren verwendet. Dabei wird jeder Term eines Dokuments bzw. einer Suchanfrage als eine Dimension (Achse) im Vektor repräsentiert. Die Anzahl der Vektorachsen ist also abhängig von der Anzahl der distinkten Terme im Dokument.
Als Vektorterm können entweder einzelne Wörter, Multiwörter („Chunks“), N-grams oder auch ganze Phrasen verwendet werden. Darüber hinaus wird jeder Term im Vektor mit einem Gewicht, welches die Relevanz des Terms in Bezug auf das Dokument wider-

spiegeln soll, versehen (siehe Tabelle 1); es handelt sich also genau genommen um Vektoren bestehend aus (Term, Gewicht)-Paaren. Wie „Relevanz“ in diesem Sinne definiert ist, hängt von der Applikation, die das Vektorraum-Modell nutzt, ab. So existiert für die Berechnung der Termgewichte eine Vielzahl an Methoden, wie beispielsweise die bekannte und häufig verwendete TF-IDF Methode (vgl. Kapitel 2.2.3).
Tabelle 1 zeigt die Repräsentation eines Dokuments im Vektorraum-Modell, wobei nicht der gesamte Vektor, sondern nur die Terme mit der höchsten Gewichtung dargestellt werden. Die Berechnung der Gewichte erfolgte nicht nach der TF-IDF Methode, sondern nach der ROBUS spezifischen Gewichtungsfunktion (vgl. Kapitel 5.1.6).

Term	*Gewicht*
web	3.05748
developer	2.56737
asp	1.83625
php	1.74986
javascript	1.56781
application	1.51373
c#	1.48317
http	1.46808
information	1.35218
sql	1.34565
work	1.31551
race	1.26924
design	1.22719
…	…

Tabelle 1: Darstellung eines Dokuments[2] als gewichteter Termvektor (Auszug der 13 Terme mit dem höchsten Gewicht)

Die Repräsentation von Dokumenten im Vektorraum-Modell bedingt unweigerlich einen Informationsverlust. So geht z.B. der relative Zusammenhang zwischen Wörtern verloren: Die Dokumente „Der Professor spricht mit seinem Studenten.“ und „Der Student

[2] Quelle: Stellenausschreibung der Internet-Plattform LinkedIn (https://www.linkedin.com/jobs2/view/10097623); Zugegriffen am 11.04.2014

spricht mit seinem Professor.“ führen zur gleichen Vektor-Repräsentation; die semantischen Unterschiede können anhand der Vektoren nicht mehr identifiziert werden. Trotz dieser Einschränkungen wird das Vektorraum-Modell im Bereich der Informationssuche von einer Vielzahl von Systemen und Forschungsprojekten verwendet.
Für eine weiterführende Erläuterung bzw. Diskussion sei an dieser Stelle auf die umfangreiche Literatur verwiesen (Manning et al. 2008), (Croft et al. 2010).
Das Vektorraum-Modell ist für die gegenständliche Arbeit von Relevanz, da es im ROBUS Verfahren an mehreren Stellen zum Einsatz kommt. Zum einen dient es als Grundlage bei der Suche nach relevanten Termvektoren in Stellenausschreibungstexten (siehe Kapitel 5.1), zum anderen basiert auch die Abbildung der zu durchsuchenden Textdokumente und Suchanfragen (siehe Kapitel 5.2) auf dem Vektorraummodell. Die konkrete Verwendung bzw. Konfiguration des Modells wird in den beiden Kapiteln detailliert beschrieben.

2.2 Gewichtung mit dem TF-IDF-Maß

Wie im vorherigen Abschnitt erläutert, wird im Vektorraum-Modell jedem Term eines Termvektors eine Gewichtung zugeordnet, die die spezifische Relevanz des Terms in Bezug auf das abgebildete Dokument repräsentiert. Terme mit einem hohen Gewicht stehen im Allgemeinen in einer engeren Beziehung zu einem Dokument, als Terme mit einer geringeren Gewichtung. Zur Berechnung von Termgewichten gibt es im Bereich der Informationssuche verschiedene Methoden. Die wichtigsten Funktionen werden nachfolgend erläutert.

2.2.1 Termfrequenz (TF)

Die rudimentärste Methode zur Ermittlung eines Gewichts besteht darin, die Häufigkeit des Vorkommens eines Terms *t* in einem Dokument *d* zu bestimmen. Dieser Wert wird in der Literatur als Termfrequenz (engl.: TF - Term Frequency) bezeichnet und folgt dem intuitiven Ansatz, wonach Wörter, die besonders häufig in einem Dokument vorkommen, auch besonders wichtig für dieses Dokument sind und es somit am besten beschreiben.
Die nachfolgende Tabelle zeigt den bereits aus Kapitel 2.1 bekannten gewichteten Termvektor, der in dieser Auflistung um die Spalte *tf (Anzahl)* erweitert wurde. Obwohl jene Wörter mit einer hohen Termfrequenz tendenziell auch hohe Gewichte aufweisen, fällt sofort auf, dass die Reihung der Gewichte nicht mit jenen der Termfrequenz übereinstimmt. So erhält z.B. das Wort mit der höchsten Termfrequenz (*t = developer; tf = 26*) nur das zweithöchste Gewicht, während das Wort *php* trotz seines kleinen *tf*-Wertes das vierthöchste Gewicht aufweist.

Term	*Gewicht*	*tf*
Web	3.05748	23
developer	2.56737	26
Asp	1.83625	13
Php	1.74986	2
javascript	1.56781	6
application	1.51373	6
c#	1.48317	3
http	1.46808	2
information	1.35218	3
Sql	1.34565	4
Work	1.31551	7
Race	1.26924	1
Design	1.22719	4
…	…	…

Tabelle 2: Erweiterte Darstellung des Termvektor-Beispiels aus Kapitel 2.1

Die Verwendung der Termfrequenz als alleiniger Gewichtungsfaktor würde die Annahme zu Grunde legen, dass alle Wörter eines Dokuments gleich wichtig (d.h. gleich aussagekräftig) sind. Dies entspricht in der Regel jedoch nicht den Tatsachen. Vielmehr werden Wörter, abhängig von der verwendeten Sprache und Domäne, unterschiedlich oft verwendet. So kommen beispielsweise die Wörter *Ausbildung* und *Qualifikation* in Stellenausschreibungstexten überdurchschnittlich oft vor, wodurch sie an Aussagekraft verlieren (eine Suche nach den beiden o.a. Wörtern in einer Job-Datenbank liefert sehr viele Treffer). Um diesem Aspekt Rechnung zu tragen, wird die Termfrequenz häufig mit der im folgenden Kapitel beschriebenen Inversen Dokumentfrequenz (IDF) kombiniert.

2.2.2 Inverse Dokumentfrequenz (IDF)

Die Dokumentfrequenz *df* gibt an, in wie vielen Dokumenten einer Sammlung *D* ein Term *t* vorkommt. Terme mit einem hohen *df* Wert sind weniger spezifisch in Bezug auf die untersuchte Textsammlung und daher weniger aussagekräftig für die Verwendung in Informationssuchsystemen. Im Gegensatz zur vorher beschriebenen Termfrequenz wirkt sich die Dokumentfrequenz daher reziprok proportional auf die Gewichtung aus.
Zur Ermittlung der Termgewichte wird die Dokumentfrequenz in Form der Inversen Dokumentfrequenz *idf* (engl.: Inverse Document Frequency) nach der Formel

$$idf_t = \log \frac{N}{df_t}$$

berechnet, wobei N die Anzahl aller Dokumente in der untersuchten Textsammlung darstellt. Dementsprechend erhalten Wörter, die in vielen Dokumenten vorkommen und daher eine geringe Aussagekraft besitzen, einen kleinen *idf* Wert, während Wörter, die nur in einigen wenigen Dokumenten vorkommen und daher eine hohe Aussagekraft besitzen, einen hohen *idf* Wert (Manning et al. 2008).

Term	*Gewicht*	*tf*	*df*	*idf*
web	3.05748	23	14294	0,70
developer	2.56737	26	18582	0,59
asp	1.83625	13	2930	1,39
php	1.74986	2	1572	1,66
javascript	1.56781	6	2144	1,52
application	1.51373	6	3216	1,35
c#	1.48317	3	1501	1,68
http	1.46808	2	4288	1,22
information	1.35218	3	5646	1,10
sql	1.34565	4	2144	1,52
work	1.31551	7	6432	1,05
race	1.26924	1	1072	1,82
design	1.22719	4	6789	1,02
…	…	…	…	…

Tabelle 3: Termvektor-Repräsentation mit Angabe der zugrundeliegenden *df* und *idf* Werte

Die Tabelle 3 zeigt erneut den gewichteten Termvektor aus Kapitel 2.1, wobei nun nicht nur die Termfrequenz *tf*, sondern auch die Dokumentfrequenz *df* und die daraus berechnete Inverse Dokumentfrequenz *idf* dargestellt werden. Die *df* und *idf* Werte wurden auf Grundlage des ROBUS Stellenausschreibungskorpus (siehe Kapitel 5.1.1) mit $N = 71468$ berechnet.

Die Verwendung der (Inversen) Dokumentfrequenz stellt eine robuste Methode zur Bewertung der Aussagekraft eines Terms in einer spezifischen Dokumentensammlung dar. Sie setzt jedoch voraus, dass dem System zur Berechnungszeit die gesamte Dokumentensammlung bekannt ist, da ansonsten der *idf* Wert nicht korrekt bestimmt werden kann. Dies ist in vielen Fällen – so auch bei ROBUS – jedoch nicht gegeben. Alternative Lösungsmöglichkeiten für dieses Problem werden in der Literatur, z.B. bei (Croft et al. 2010) oder (Joel et al. 2006), beschrieben. Die konkrete Lösung für ROBUS wird in Kapitel 5.1.6 erläutert.

2.2.3 TF-IDF basierte Termgewichtung

Mit Hilfe der Termfrequenz *tf* und der Inversen Dokumentfrequenz *idf* kann für jeden Term *t* in jedem Dokument *d* innerhalb einer Dokumentensammlung *D* das spezifische Termgewicht anhand der Formel

$$tf\_idf_{t,d} = tf_{t,d} * idf_t$$

berechnet werden. Daraus ergibt sich, dass jene Wörter (Terme) die höchsten Gewichtswerte erhalten, die in möglichst wenigen Dokumenten möglichst oft vorkommen. Wörter, die innerhalb einer Textsammlung in sehr vielen Dokumenten präsent sind (wie z.B. *Ausbildung* oder *Qualifikation* in einer Sammlung von Stellenausschreibungen) erhalten einen geringen *idf* Wert und damit ein insgesamt geringes Gewicht im Termvektor. Somit werden Wörter, die für Informationssuchsysteme nur eine geringe Relevanz in Bezug auf eine Suchanfrage besitzen, weiter nach hinten gereiht.
Als Beispiel sei an dieser Stelle der Term *developer* aus Tabelle 4 erwähnt. Obwohl er die höchste Termfrequenz im Vektor aufweist (*tf = 26*) erhält er auf Grund des geringen *idf* Wertes nur das dritthöchste Gewicht, während der Term *asp* mit nur halb so vielen Vorkommnissen im Dokument (*tf = 13*) als wichtigster (d.h. als höchstgewichteter) Term im Vektor geführt wird.

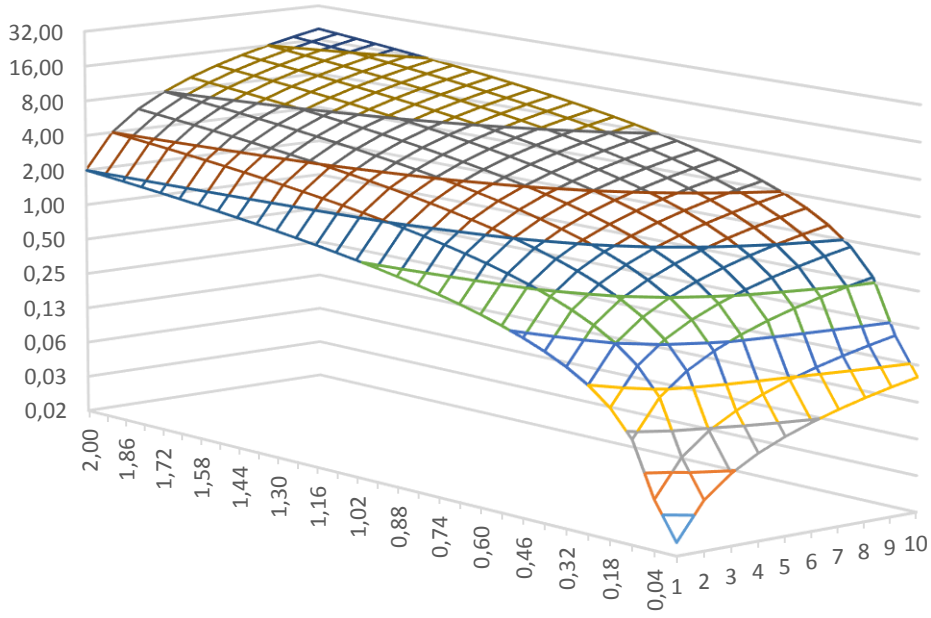

Abbildung 1: Verlauf des *tf-idf* Gewichtes in Abhängigkeit von der Termfrequenz (*tf* = 1 .. 10) und der Dokumentfrequenz (*idf* = 2 .. 0)

Abbildung 1 illustriert die Veränderung der *tf-idf* Gewichtung in Abhängigkeit der beiden zugrunde liegenden Parameter Termfrequenz und Dokumentfrequenz. Ersterer wird auf der x-Achse mit einem Werte-Bereich von *tf = 1 .. 10*, zweiterer auf der y-Achse mit einem Wertebereich von *idf = 2 .. 0* dargestellt. Die jeweiligen *tf-idf* Werte sind im Diagramm auf der z-Achse verdeutlicht. Wie aus der Abbildung hervorgeht, erlangen jene Wörter die höchste Relevanz, die über eine große Häufigkeit innerhalb eines Dokuments sowie einer kleinen Streuung im Korpus und damit über eine geringe Dokumentfrequenz verfügen.

Das *tf-idf* Maß zählt zu den meistverwendeten Gewichtungsmaßen im Bereich der Informationssuche und kommt in zahlreichen Systemen und Forschungsprojekten zum Einsatz. Für weiterführende Informationen zum *tf-idf* Maß selbst, sowie zu alternativen Gewichtungsmethoden sei an dieser Stelle auf die umfangreiche Literatur verwiesen. Einen sehr guten Überblick über *tf-idf* Berechnung selbst sowie zu der damit verbundenen Thematik der Normalisierung liefern (Manning et al. 2008) in Kapitel 6.3 und 6.4. Informationen zur Anwendung der *tf-idf* Gewichtung in aktuellen Suchmaschinen liefern z.B

(Croft et al. 2010). Eine alternative Methode zur Gewichtung von Termen ohne vollständige Kenntnis der Korpus-Kennzahlen (*df* und *N*) bieten beispielsweise (Joel et al. 2006).

Term	*Gewicht*	*tf*	*df*	*idf*	*tf-idf*
web	3.05748	23	14294	0,70	16,08
developer	2.56737	26	18582	0,59	15,21
asp	1.83625	13	2930	1,39	18,03
php	1.74986	2	1572	1,66	3,32
javascript	1.56781	6	2144	1,52	9,14
application	1.51373	6	3216	1,35	8,08
c#	1.48317	3	1501	1,68	5,03
http	1.46808	2	4288	1,22	2,44
information	1.35218	3	5646	1,10	3,31
sql	1.34565	4	2144	1,52	6,09
work	1.31551	7	6432	1,05	7,32
race	1.26924	1	1072	1,82	1,82
design	1.22719	4	6789	1,02	4,09
…	…	…	…	…	…

Tabelle 4: Termvektor-Repräsentation mit Gegenüberstellung des ROBUS Gewichts mit den korrespondierenden *tf-idf* Werten

Tabelle 4 zeigt einen Auszug des ursprünglichen Termvektors aus Kapitel 2.1, wobei die Darstellung um die Spalten *tf, df, idf* und *tf-idf* erweitert wurde. Vergleicht man die beiden Spalten *Gewicht* (die Gewichtung, die vom ROBUS Verfahren für die Termvektoren berechnet wird) und *tf-idf*, ist zu erkennen, dass nicht nur die Absolutwerte, sondern auch die relative Reihung unterschiedlich ausfallen.

Der Grund dafür liegt darin, dass ROBUS zwar das grundlegende Prinzip des *tf-idf* Maßes verwendet, dieses jedoch um mehrere domänen- und anwendungsspezifische Anpassungen erweitert bzw. ergänzt. So kommt beispielsweise ein domänenspezifischer Thesaurus zum Einsatz, der eine Zuordnung von Termen zu Fachbereichen ermöglicht. Des Weiteren wird das untersuchte Dokument in mehrere Zonen (Abschnitte) unterteilt. Für jede Zone und jeden Term wird ein separates Gewicht, abhängig von seinem *tf* Wert sowie der Position des Terms im Text, ermittelt und anschließend entsprechend der Zonengewichtung zusammengeführt.

Eine detaillierte Beschreibung der Gewichtungsmethode von ROBUS findet sich in Kapitel 5.1.6.

2.3 Ähnlichkeiten im Vektorraum-Modell

Wie bereits eingangs erläutert, werden Dokumente im Vektorraum-Modell in Form von gewichteten Termvektoren repräsentiert, wobei jeder Term *t* eines Dokuments *d* eine Dimension im Vektor $\vec{V}_d$ darstellt und als solches ein spezifisches Gewicht aufweist. Im Allgemeinen verläuft das Termgewicht proportional zur Relevanz von *t* in Bezug auf *d*, wobei ein Gewicht von 0 bedeutet, dass der Term *t* nicht in *d* enthalten ist.
Im Bereich von Informationssuchsystemen können nicht nur Dokumente, sondern auch Suchanfragen (engl.: Query) im Vektorraum-Modell abgebildet werden. Die Vorgehensweise ist dabei ident mit jener der Dokumentrepräsentation: Jeder Suchbegriff wird als eine Dimension im Vektor dargestellt. Die Gewichtung wird vom Suchsystem im Zuge der Erstellung der Suchanfrage ermittelt und ist vom jeweiligen System abhängig.
Die Repräsentation von Dokumenten als gewichtete Vektoren ermöglicht nicht nur eine quantitative Bestimmung der Ähnlichkeit von Inhalten unterschiedlicher Dokumente auf Basis der Ähnlichkeit ihrer korrespondierenden Termvektoren. In gleicher Weise kann auch die Ähnlichkeit zwischen einer Suchanfrage und einem Dokument berechnet werden. Diese Bewertungen finden sich in der Literatur zumeist unter der englischen Bezeichnung *Search Score* bzw. *Scoring*. Zur Berechnung der Ähnlichkeit zwischen Vektoren gibt es verschiedene mathematischen Funktionen. Im Bereich der Informationssuche gilt die Methode der Kosinus-Ähnlichkeit als das Standardverfahren. Wie der Name schon verrät, wird dabei der Kosinus des Winkels zwischen den beiden untersuchten Vektoren ermittelt. Dies bietet im Vergleich zu anderen Verfahren den Vorteil, dass der Ähnlichkeitswert nicht von der Länge des Vektors und folglich nicht von der Länge des Dokuments bzw. der Suchanfrage abhängt. Für Informationssuchsysteme stellt das einen

wesentlichen Faktor dar, da Suchanfragen in der Regel sehr viel kürzer sind, als die durchsuchten Dokumente.

$$sim(d,q) = cos\,\theta = \frac{\vec{V}_d * \vec{V}_q}{|\vec{V}_d| * |\vec{V}_q|}$$

Die Ähnlichkeit *sim(d,q)* zwischen einem Dokument *d* und einer Suchanfrage *q* errechnet sich anhand obenstehender Formel. Dabei stellt $\vec{V}_d$ den Termvektor des durchsuchten Dokuments *d* und $\vec{V}_q$ den Termvektor der Suchanfrage *q* dar. Aus dieser Formel ergibt sich ein proportionaler Zusammenhang zwischen dem Kosinuswert und der Ähnlichkeit von Dokumenten bzw. Suchanfragen: Umso höher der Kosinuswert, desto ähnlicher die Dokumente. Ein Ähnlichkeitswert von 1 bedeutet eine völlige Übereinstimmung (die beiden Vektoren sind ident), ein Wert von 0 bedeutet im Umkehrschloss den größtmöglichen Unterschied zwischen zwei Vektoren. Negative Werte sind nicht möglich, da Terme keine negativen Gewichtungen haben.

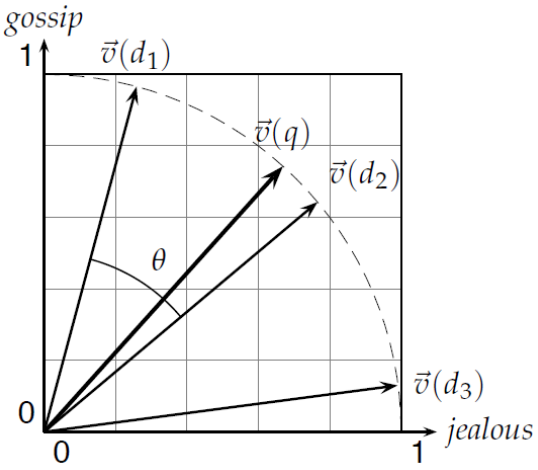

Abbildung 2: Darstellung der Kosinus-Ähnlichkeit zwischen den Termvektoren von Dokumenten *d1, d2 und d3* sowie der Suchanfrage *q*. (Manning et al. 2008)

Abbildung 2 beinhaltet eine grafische Darstellung der Kosinus-Ähnlichkeit anhand mehrerer Termvektoren. Alle enthaltenen Vektoren weisen zwei Dimensionen (*gossip* und *jealous*) sowie normalisierte Gewichtungswerte auf. Aus der Illustration lässt sich ableiten, dass sich die Vektoren von *q* und *d2* am ähnlichsten sind, während die Vektoren von *d1* und *d3* die geringste Ähnlichkeit aufweisen.

Informationssuchsysteme verwenden genau dieses Verfahren, um bei einer Suchanfrage $\vec{V}_q$ jene Dokumente in einer Textsammlung *D* zu finden, die dem Vektor der Suchanfrage am ähnlichsten sind. Dieses Vorgehen basiert auf der grundlegenden Annahme, dass die ähnlichsten Dokumente auch genau jene Dokumente sind, die das Informationsbedürfnis in Bezug auf die Suchanfrage am besten befriedigen. Das Kosinus-Verfahren gilt auch im Bereich der Informationssuche als die Standardmethode zur Berechnung von Ähnlichkeiten im Vektorraum-Modell. Daher sei für nähere Informationen auf die ausführliche bestehende Literatur, wie z.B. (Croft et al. 2010), (Manning & Schuetze 1999) oder (Chim & Deng 2007) verwiesen.

Das ROBUS System verwendet das Kosinus-Verfahren nicht nur zur Bestimmung der Ähnlichkeit zwischen Suchanfrage und durchsuchtem Dokument. Vielmehr bestimmt ROBUS anhand der Kosinus-Ähnlichkeit auch die Relevanz eines Dokuments für ein spezifisches Rollenprofil. Rollenprofile in ROBUS sind gewichtete Termvektoren, die eine bestimmte Unternehmensrolle spezifizieren und damit eine kontextsensitive Adaption der Suchergebnisse unabhängig von der eingegebenen Suchanfrage erlauben. Die konkrete Methodik sowie der Einsatz der Kosinus-Ähnlichkeit werden in Kapitel 5.2.1 erläutert.

2.4 Das probabilistische Relevanzmodell

Das BM25 Verfahren zählt zur Gruppe der probabilistischen Informationssuchsysteme und findet heute in verschiedensten Systemen aus dem Bereich der Informationssuche Anwendung. Wie auch das Vektorraum-Modell, wird BM25 zur Bestimmung der Relevanz von Dokumenten in Bezug auf Suchanfragen verwendet. Innerhalb der wissenschaftlichen Forschung im Bereich der Informationssuche gilt das Verfahren heute als die fortschrittlichste Methode zur Reihung von Suchergebnissen (Perez-Iglesias et al. 2009; Trotman & Keeler 2011; Garrido et al. 2010; Clinchant 2012; Robertson & Zaragoza 2009; Robertson 1997). In der Literatur findet sich häufig auch die Bezeichnung Okapi BM25, was darauf zurückzuführen ist, dass das Okapi[3] Informationssuchsystem das erste war, das diese Funktion implementierte. (Robertson 1997)

Der BM25 Algorithmus basiert auf dem Konzept des „Probabilistic Relevance Framework“ (PRF), welches bereits in den 1970er und 1980er Jahren entwickelt wurde. Das PRF Modell ermittelt eine Wahrscheinlichkeit, die darüber Auskunft gibt, ob bzw. wie ein Dokument *d* für eine Suchanfrage *q* relevant ist. Dieser Wahrscheinlichkeitsfunktion liegt die Annahme zugrunde, dass eine Teilmenge an Dokumenten *R* aus der Menger aller Dokumente *D* vom Benutzer als relevant in Bezug auf eine bestimmte Suchanfrage erachtet und somit als Suchergebnis zurück geliefert werden sollte. Ausgehend von diesen grundsätzlichen Überlegungen wurden von Robertson et al mehrere konkrete Implementierungen des Modells, wie etwa das „Binary Independence Model“, das „Eliteness Model“ oder das „2-Poisson Model“ entwickelt. Das am weitetesten verbreitete Modell ist jedoch ohne Zweifel die „BM25 Term-weighting and Document-scoring Function“ (Robertson & Zaragoza 2009).

[3] Okapi ... Familie von Informationssuchsystemen, die in den 1990er Jahren unter der Leitung von Stephen Robertson am University College London entwickelt wurden: http://www.soi.city.ac.uk/~ser/

$$RSV_d = \sum_{t \in q} \left[\log \frac{N}{\mathrm{df}_t} \right] \cdot \frac{(k_1 + 1)\mathrm{tf}_{td}}{k_1((1-b) + b \times (L_d / L_{\mathrm{ave}})) + \mathrm{tf}_{td}} \cdot \frac{(k_3 + 1)\mathrm{tf}_{tq}}{k_3 + \mathrm{tf}_{tq}}$$

Gleichung 1: Berechnung des Gewichtungsschemas im BM25 Modell nach (Manning et al. 2008)

Die obige Gleichung zeigt eine der gebräuchlichsten Formeln zur Gewichtungsberechnung im BM25 Modell. Dabei wird für jeden Term *t* in einer Suchanfrage *q* die Inverse Dokumentfrequenz (vgl. Kapitel 2.2.2) mit zwei weiteren Werten multipliziert. Ersterer ist abhängig von der Termfrequenz tf_{td} des Terms *t* im durchsuchten Dokument *d* sowie den beiden konfigurierbaren Parametern k_1 (zur Bestimmung des Einflusses von tf_{td} auf den Gesamtwert) und *b* (Bestimmung der Bedeutung der Längennormalisierung). Die Normalisierung der Dokumentenlänge wird durch die beiden Parameter L_d (Länge des durchsuchten Dokuments *d*) sowie L_{ave} (durchschnittliche Länge aller Dokumente) ermittelt, kann jedoch durch die Angabe von $b = 0$ deaktiviert werden.
Auf eine vollständige Herleitung der BM25 Funktion sowie einer tiefgehenden Diskussion wird an dieser Stelle verzichtet und stattdessen auf die bestehende Literatur verwiesen. Eine gute Einführung in probabilistische Suchmodelle im Allgemeinen und BM25 im Speziellen geben z.B. (Croft et al. 2010), (Robertson 1997) und (Manning et al. 2008). Eine frei verfügbare Implementierung von BM25 auf Basis der Open Source Plattform Apache Lucene[4] findet sich in (Perez-Iglesias et al. 2009).
Das BM25 Modell ist für die gegenständliche Arbeit nicht zuletzt deshalb von Interesse, da es bei der Evaluation des ROBUS Verfahrens als sogenannte Baseline verwendet wird. Dabei werden die Ergebnisse des ROBUS Verfahrens jenen des Baseline Systems gegenübergestellt und der Mehrwert von ROBUS gemessen bzw. beurteilt. Das detaillierte Testverfahren inklusive aller verwendeten Komponenten und Konfigurationen wird im Kapitel 0 ausführlich diskutiert.

[4] Apache Lucene ist eine Java-basierte Open Source Plattform, die umfangreiche Funktionen im Bereich der textuellen Informationssuche zur Verfügung stellt. Die Software und Dokumentation können von http://lucene.apache.org/core/ bezogen werden.

3 Evaluation von Suchsystemen

Die Durchführung einer umfassenden Evaluation eines Suchsystems ist eine unverzichtbare Grundlage für eine aussagekräftige Beurteilung der Suchergebnisse beziehungsweise der Funktionsweise des Systems an sich. Sie ist außerdem eine wichtige Voraussetzung für jede Weiterentwicklung und Optimierungsmaßnahme, da erst mit Hilfe einer Evaluation die konkreten Auswirkungen einer Veränderung auf das System festgestellt (gemessen) werden können.

Das folgende Kapitel beschreibt mögliche Ansätze und Verfahren zur Evaluation von Suchsystemen sowie die dazu verwendeten Komponenten und Bewertungsmetriken. Ein besonderer Schwerpunkt wird dabei auf die Evaluationsmöglichkeiten für personalisierte Suchstrategien gelegt.

3.1 Einleitung

Bei der Evaluation von Suchsystemen unterscheidet man prinzipiell zwischen der Beurteilung der Effektivität und der Beurteilung der Effizienz eines untersuchten Systems. Mit der Effektivität wird bewertet, in welchem Maße das System dazu in der Lage ist, die „richtigen" Informationen – das sind jene Dokumente, die für eine bestimmte Suchanfrage als relevant definiert wurden – in der richtigen Reihenfolge zu finden. Die Effizienz hingegen gibt Auskunft darüber, in welcher Zeit beziehungsweise mit welchen Systemressourcen die Suchanfragen bearbeitet wurden. Beide Faktoren, sowohl die Effektivität als auch die Effizienz, werden von einer Reihe von Parametern wie beispielsweise der Benutzeroberfläche oder den Interaktionsmöglichkeiten beeinflusst. Daher ist es für jede Evaluationsdurchführung von entscheidender Bedeutung, alle relevanten Rahmenbedingungen genau festzulegen und zu kontrollieren (Croft et al. 2010).
Im Allgemeinen wird bei der Entwicklung von Suchsystemen in der wissenschaftlichen Community der Schwerpunkt zuerst auf den Effektivitätsfaktor gelegt. Erst wenn sich gezeigt hat, dass die neue Methode zu einer Verbesserung des Suchvorgangs beitragen kann und sich eine Weiterentwicklung lohnt, wird an der Optimierung der Effizienz gearbeitet. Dies bedeutet nicht, dass Entwicklungsarbeit im Bereich der Effizienz generell weniger wichtig ist als im Bereich der Effektivität. Umgekehrt kann es auch vorkommen, dass Maßnahmen, die zu einer Verbesserung der Effektivität beitragen, bewusst nicht in das Produktivsystem integriert werden, wenn sie so hohe Effizienzeinbußen verursachen, dass die – durch die Maßnahme erreichte - Steigerung der Effektivität nicht mehr gerechtfertigt ist (Croft et al. 2010). Die Autoren (Croft et al. 2010) sehen im Kontext der Informationssuche aber dennoch keinen Konflikt zwischen den beiden Faktoren, da derzeit keine Suchmethode bekannt ist, deren Einsatz ausschließlich aus Effizienzgründen ausgeschlossen wird.

3.2 Testkorpora als Evaluationsgrundlage

Unter einem Korpus versteht man im Kontext der Sprachwissenschaften nach Definition von Bussmann eine „endliche Menge von konkreten sprachlichen Äußerungen, die als empirische Grundlage für sprachwissenschaftliche Untersuchungen dienen. Stellenwert und Beschaffenheit des C[orpus] hängen weitgehend von den je spezifischen Fragestellungen und methodischen Voraussetzungen des theoretischen Rahmens der Untersuchung ab [...]“ (Bubenhofer 2011).

Für die Anwendung im Bereich der Informationssuche beinhalten Textkorpora neben den textuellen Dokumenten noch Suchanfragen (engl.: Query) und Relevanzbeurteilungen (engl.: Relevance judgement), wobei eine Relevanzbeurteilung definiert, welche Dokumente des Korpus' relevant für eine bestimmte Suchanfrage sind. Der Einsatz solcher Korpora zur Evaluation von Suchsystemen geht zurück bis in die 1960er Jahre. Zu dieser Zeit wurden im Zuge der sogenannten „Cranfield Experimente“ die ersten großflächigen Untersuchungen in diesem Gebiet an der gleichnamigen Universität unter der Leitung von Cyril W. Cleverdon durchgeführt (Cleverdon 1991).

Ziel dieser Experimente war es, die Effektivität von Suchsystemen mittels verbesserter Indizierungssprachen und -methoden zu steigern. Die Experimente wurden auf Basis eines selbst erstellten Korpus mit den oben angeführten Inhalten (Dokumente, Suchanfragen und Relevanzbeurteilungen) und unter genau kontrollierten Testbedingungen durchgeführt. Zur quantitativen Beurteilung der Ergebnisse wurden die Relevanzmaße Genauigkeit (engl.: Precision) und Trefferquote (engl.: Recall) herangezogen. Die Cranfield Experimente wurden schnell zum Musterbeispiel für die wissenschaftliche Suchsystem-Evaluation und werden heute oft als „Beginn der modernen Ära der Computer-basierten Evaluation von Suchsystemen“ bezeichnet (Charles 2001).

Auf Grund der sorgfältig definierten Testbedingungen sowie der aussagekräftigen und gut vergleichbaren quantitativen Bewertungsmaße genießt die Cranfield Methode in der wissenschaftlichen Gemeinde auch heute noch hohes Ansehen, wenngleich in der Vergangenheit mehrere Schwachpunkte aufgezeigt wurden. So kritisiert Hildreth in (Charles 2001), dass „sich das Cranfield Modell fast ausschließlich auf das attraktive aber schwierige Konzept der Relevanz stützt“ und eine rein system-orientierte Evaluation darstellt.

Des Weiteren bemängelt der Autor, dass die beiden dem Modell zugrunde liegenden fundamentalen Annahmen

(1) Anwender möchten nur jene Dokumente als Suchergebnis geliefert bekommen, die relevant in Bezug auf die eingegeben Suchanfrage sind, aber nicht solche, die nicht relevant für die Suchanfrage sind und

(2) die Relevanz eines Dokuments für eine bestimmte Suchanfrage ist eine objektive und eindeutige Eigenschaft des Dokuments

in mehreren unterschiedlichen Untersuchungen widerlegt wurden und mittlerweile nicht mehr als gültig angesehen werden können. Dennoch zählt die Cranfield Methode bis heute zu einer der populärsten und am weitesten verbreiteten Ansätze und „hat in den letzten 30 Jahren als Grundlage für den Großteil der durchgeführten Evaluationen im Bereich der Informationssuche gedient“ (Charles 2001). Dies erklärt Hildreth vor allem damit, dass die Benutzer/innen keinen direkten Einfluss auf das Experiment nehmen können und dass sämtliche Anwender-bezogenen Faktoren ignoriert werden können.

3.3 Die TREC Reihe

Im Laufe der Jahre wurden zahlreiche Korpora nach dem Vorbild von Cranfield entwickelt. Die wohl bekanntesten und am häufigsten verwendeten sind jene der TREC (Text Retrieval Conference) Reihe. Diese Konferenz wird seit 1992 jährlich veranstaltet und wurde mit dem Ziel ins Leben gerufen, die Forschungsaktivitäten auf dem Gebiet der Informationssuche durch die Bereitstellung von umfangreichen Testdatensätzen zu unterstützen (NIST 2010).
Seither haben mehr als 250 Forschungsgruppen aus über 20 Ländern an TREC Konferenzen teilgenommen. Im Zuge dessen wurden tausende Experimente durchgeführt und hun-

derte Publikationen veröffentlicht. Damit gilt TREC in der wissenschaftlichen Community unumstritten als die einflussreichste Veranstaltungsreihe in diesem Bereich. Für die einzelnen TREC Konferenzen wurde eine Vielzahl unterschiedlicher Korpora generiert, die neben der traditionellen ad-hoc Suche auch für neuere beziehungsweise speziellere Einsatzszenarien wie beispielsweise sprachübergreifende Suche, sprachliche Suche oder Frage-Beantwortungssysteme (engl.: Question Answering) gedacht sind (Voorhees 2005).

Wie bereits erwähnt, basiert TREC auf der Cranfield Methode. Zu den grundlegenden Komponenten jedes TREC Korpus gehören daher neben den eigentlichen Dokumenten eine Menge von Themen (engl.: Topics), die die jeweiligen Informationsbedürfnisse repräsentieren und aus denen in weiterer Folge die Suchanfragen (engl. Query) für das evaluierte System abgeleitet werden, sowie eine Menge von Relevanzbeurteilungen (engl. Relevance Judgment), die festlegen, welche Dokumente für welche Themen relevant sind und somit vom Suchsystem gefunden werden sollten.

Diese Testdaten werden für jede TREC Konferenz vom amerikanischen Hauptsponsor NIST (National Institute of Standards and Technology) bereit gestellt. Die durchschnittliche Größe sowie die Anzahl von Dokumenten variiert sehr stark. So beinhaltet beispielsweise der GOV2 Datensatz (Terabyte Track) über 25 Millionen Dokumente und weist eine Gesamtgröße von 426GB auf, während der AP (Associated Press) Datensatz mit knapp 243.000 Dokumenten und einer Größe von ca. 700MB deutlich kleiner ist. Im Schnitt beinhalten die TREC Datensätze ca. 800.000 Dokumente (Voorhees 2005; Croft et al. 2010).

Zusätzlich zu den Dokumenten enthält jede TREC Reihe 50 neue Themen. Das Format der Themen unterscheidet sich von Reihe zu Reihe enthält aber zumindest eine textuelle Beschreibung sowie einen Titel (Anfrage). Die Themen werden von Mitarbeitern des NIST erarbeitet und dienen als Grundlage für die Suchanfragen der teilnehmenden Gruppen und Systeme. Als Ergebnis liefern die TREC Teilnehmer für jedes Thema eine Liste der höchstgereihten Dokumente zurück an NIST. Die Relevanzbewertungen für jedes TREC Korpus werden manuell erstellt. Da die Dokumentsammlungen mit durchschnittlich 800.000 Datensätzen zu groß sind, um jedes einzelne Dokument hinsichtlich seiner Relevanz für ein Thema zu bewerten, wurde eine eigene Methode namens POOLING

entwickelt. Dabei werden die jeweils 100 höchstgereihten Suchergebnisse von jedem teilnehmenden Suchsystem in einen gemeinsamen Topf (engl.: Pool) gegeben. Dokumente, die von mehreren Suchsystemen als Ergebnis übermittelt wurden, werden nur ein Mal zum Topf hinzugefügt. Anschließend wird für jedes im Topf befindliche Dokument eine manuelle Relevanzbewertung erstellt. Dieses Vorgehen führt zu einer signifikanten Reduktion der zu bewertenden Dokumente und ermöglicht so überhaupt erst die Durchführung von manuellen Relevanzbewertungen.
Sobald alle Relevanzbewertungen für sämtliche Themen vorliegen, werden die Suchergebnisse der einzelnen Teilnehmersysteme von NIST auf Basis dieser Relevanzbewertungen evaluiert. Während der ersten zwei TREC Reihen gab es je zwei Aufgaben (engl. Task): der Ad-hoc Task repräsentierte die traditionelle Suchaufgabe, bei der eine bekannte Menge von Dokumenten nach unterschiedlichen Aspekten durchsucht wird. Im Gegensatz dazu galt es beim Routing Task stetig neue Dokumente nach vorgegebenen, gleichbleibenden Themen zu durchsuchen. Diese Aufgabe sollte die Anforderungen von Profilerstellungsdiensten (z.B. Pressespiegel, Bibliotheksprofile, usw.) widerspiegeln. Ab dem dritten Jahr (TREC-3) wurden zusätzliche Aufgaben (sogenannte Tracks) und Testdatensätze eingeführt, um ein breiteres Spektrum an Anforderungen abdecken und sich damit einem größerem Forschungspublikum öffnen zu können (Voorhees 2005).

```
<topic number="153" type="faceted">
  <query>pocono</query>
  <description>
    Find general information on tourist activities in
    Pennsylvania's Pocono Mountains.
  </description>
  <subtopic number="1" type="inf">
    Find general information on tourist activities in
    Pennsylvania's Pocono Mountains.
  </subtopic>
  <subtopic number="2" type="nav">
    Find a map showing lodgings in the Poconos PA region.
  </subtopic>
  <subtopic number="3" type="inf">
    Find information about the Pocono Raceway in Pennsylvania.
  </subtopic>
  <subtopic number="4" type="inf">
    Find information about the Split Rock Resort in the Poconos.
  </subtopic>
</topic>
<topic number="154" type="faceted">
  <query>figs</query>
  <description>
    Find information on nutritional or health benefits of figs.
  </description>
  <subtopic number="1" type="inf">
    Find information on nutritional or health benefits of figs.
  </subtopic>
  <subtopic number="2" type="nav">Find recipes that use
  figs.</subtopic>
  <subtopic number="3" type="inf">
    Find information on the different varieties of figs.
  </subtopic>
  <subtopic number="4" type="inf">Find information on growing
  figs.</subtopic>
</topic>
```

Abbildung 3: Beispiel für Themen (Topics) aus dem TREC 2012 Web Track (NIST 2012)

Wie bereits erwähnt, war eines der Hauptziele von TREC eine standardisierte Evaluierungsinfrastruktur für Informationssuchsysteme bereit zu stellen. Denn nur auf Basis einer solchen standardisierten Infrastruktur können Ergebnisse unterschiedlicher Systeme objektiv beurteilt und miteinander verglichen werden. Neben den oben erläuterten Testdatensätzen ist eine weitere wesentliche Komponente der TREC Infrastruktur das Evaluationsprogramm TREC_EVAL. Dieses Programm ist ebenfalls frei verfügbar[5] und wurde von Chris Buckley entwickelt. Es beinhaltet eine Implementierung von über 100 Evalu-

[5] http://trec.nist.gov/trec_eval/

ierungsmaßen und wird von der NIST zur offiziellen Bewertung der von den TREC Teilnehmern eingesendeten Suchergebnisse verwendet. Zur Beurteilung der Systemeffektivität haben sich im Laufe der Jahre eine kleine Anzahl von Kriterien als Quasi-Standard etabliert, dazu zählen (1) Recall-Precision-Graph, (2) Mean Average Precision und (3) Precision At Ten Retrieved Documents. Die nachfolgenden Kapitel beinhalten eine ausführliche Beschreibung dieser Kriterien (Voorhees 2005).

3.4 Implizite Relevanzbewertungen

Die Generierung von Korpora und Testdatensätzen kann mit sehr hohem Aufwand bzw. signifikanten Kosten verbunden sein. Vor allem die Definition von Relevanzbewertungen stellt in diesem Sinne oft ein Problem dar, da eine vollständige manuelle Erfassung solcher Daten bei größeren Dokumentensammlungen nicht mehr möglich ist. Speziell entwickelte Technologien zur Reduktion des Arbeitsumfangs, wie beispielsweise die Bildung von repräsentativen Teilmengen bei der von TREC verwendeten POOLING Methode (Details siehe Kapitel 3.3), bieten zwar eine Lösung des Problems, sehen sich aber gleichzeitig großer Kritik ausgesetzt. So wird zum Beispiel bei der POOLING Methode bemängelt, dass Systeme, deren Ergebnisse nicht in den Pool aufgenommen wurden, generell benachteiligt sind. Grund dafür ist, dass die Suchergebnisse solcher Systeme als nicht korrekt beurteilt werden, auch wenn sie für eine Suchanfrage relevant sind, weil für diese Dokumente keine Relevanzbeurteilungen durchgeführt wurden (Buckley et al. 2006).

Ein alternativer Ansatz zur Generierung von Relevanzbewertungen ist die Auswertung von impliziter Benutzerresonanz (engl.: Implicit Feedback). Dabei wird davon ausgegangen, dass bestimmte Benutzeraktionen - wie beispielsweise das Klicken und Betrachten eines Dokuments in der Suchergebnisliste - als Relevanzbewertung interpretiert werden können. Diese Vorgehensweise ermöglicht eine signifikante Reduktion des Aufwands bei

der Erstellung von Relevanzbewertungen, weist aber ebenfalls mehrere Schwachstellen auf, die beim Einsatz dieser Methode auf jeden Fall berücksichtigt werden sollten. Das nachfolgende Kapitel beschreibt die grundlegende Methode der Nutzung von implizitem Feedback zur Evaluation von Suchsystemen. Weiters werden die Vor- und Nachteile solcher Relevanzbewertungen gegenüber Bewertungen, die mittels POOLING erstellt wurden, dokumentiert.

Interaktionen zwischen Anwendern und Suchmaschinen werden von den Systemen in Form von sogenannten „Logging" Daten protokolliert. Diese Logging Daten (oft auch als „Query Logs" bezeichnet) stellen mittlerweile für den Großteil der Forschungsgemeinde eine äußert wichtige Ressource bei der Entwicklung und Evaluierung von Suchsystemen dar. Vom Autor wurde eine Untersuchung von Publikationen der CIKM (ACM Conference on Information and Knowledge Management) Konferenzreihe im Zeitraum von 2006 bis 2010 durchgeführt. Die CIKM wurde 1992 ins Leben gerufen und richtet sich vor allem an Forschungsgruppen aus den Bereichen Wissensmanagement und Informationssuche. Übergeordnetes Ziel der Konferenzreihe ist es, „herausfordernde Probleme bei der Entwicklung zukünftiger Wissens- und Informationssysteme zu identifizieren und zukünftige Forschungsrichtungen durch die Veröffentlichung von qualitativ hochwertigen, angewandten und theoretischen Forschungsergebnissen mit zu gestalten" (CIKM 2013). Die CIKM gilt als eine der qualitativ hochwertigsten und zukunftsweisenden Konferenzreihen auf diesem Gebiet.

Die oben angeführte Untersuchung ergab, dass von den 42 analysierten Publikationen, die eine Beschreibung der Evaluationsmethode beinhalteten, insgesamt 19 mit Logging Daten arbeiteten. Wie in Tabelle 5 weiters ersichtlich, kann ein konstanter Anstieg beim Einsatz von Logging Daten verzeichnet werden. So wurden im Jahr 2010 bereits sieben von zehn Arbeiten auf Basis von impliziten Relevanzbewertungen aus Query Logs durchgeführt.

Jahr	Cranfield Methode	Query Log Daten	Individuelle Experimente
2006	2	2	4
2007	4	2	1
2008	6	2	0
2009	1	6	2
2010	1	7	2
Summe	**14**	**19**	**9**

Tabelle 5: Einsatz von Logging Daten zur Evaluation von Suchsystemen nimmt zu

Protokollierte Query Log Daten von Suchsystemen können enorme Mengen an Informationen liefern und Auskunft darüber geben, wie die Anwender/innen mit den bereitgestellten Suchergebnissen interagieren. Entwickler, die Zugang zu Query Logs von großen öffentlichen Internetsuchmaschinen wie beispielsweise Google oder Microsoft Bing haben, können so auf mehrere Millionen Datensätze zugreifen und für ihre Testzwecke nutzen. Der Einsatz eines solchen Datenbestandes ermöglicht weitaus umfangreichere und realistischere Evaluationen von Informationssuchsystemen als mit den vergleichsweise eingeschränkten Testdatensätzen der TREC Reihe (Croft et al. 2010). Es bestehen aber auch mehrere Nachteile beziehungsweise Einschränkungen bei der Verwendung von impliziten Relevanzbewertungen. Dazu zählen laut (Croft et al. 2010) die folgenden drei Punkte:

(1) Ein großes Hindernis besteht in der generellen Verfügbarkeit von Query Log Datensätzen. Während die TREC Korpora und deren Dokumentation für alle Forschungsgruppen frei zugänglich sind, handelt es sich bei Logging Daten um proprietäre Aufzeichnungen, die nur einem eingeschränkten Benutzerkreis (oft ausschließlich dem veröffentlichenden Autor/den veröffentlichenden Autoren) zur Verfügung stehen. Das Vergleichen und Nachvollziehen von Forschungsergebnissen, und somit auch eine zielführende Weiterentwicklung durch andere Forschungsgruppen, wird dadurch erheblich erschwert oder sogar unmöglich gemacht.

(2) Ein weiterer Nachteil von impliziten Relevanzbewertungen liegt in der geringeren Präzision der Bewertungsdaten. Während explizite Relevanzbewertungen von geschulten und unabhängigen Personen in kontrollierten Laborumgebungen erstellt werden, entstehen implizite Bewertungen durch die automatisierte Aufzeichnung von Benutzerinteraktionen. Tiefergehende Informationen zu den Personen oder den protokollierten Interaktionen sind in aller Regel nicht verfügbar und können somit auch nicht bei der Erstellung der Relevanzbewertungen einfließen. Beim expliziten Feedback ist sichergestellt, dass ausschließlich relevante Dokumente als solche definiert werden. Im Gegensatz dazu muss bei impliziten Feedback Daten auf eine ausreichende Korrektheit der zugrunde gelegten Annahmen (z.B. „wenn der Benutzer ein Dokument aus dem Suchergebnis anklickt und betrachtet, gilt es als relevant für die gegebene Suchanfrage“) vertraut werden.

(3) Ein besonders schwieriges – da sehr vielschichtiges – Problem beim Einsatz von impliziten Feedback Daten ist das Thema Privatsphäre und Datenschutz. Es existiert eine Vielzahl von unterschiedlichen Rahmenbedingungen, die wiederum von einer Reihe von Einflussparametern abhängig sind. Daher ist es notwendig, Query Log Informationen zu anonymisieren bevor diese geteilt, veröffentlicht oder publiziert werden können, auch wenn dadurch der Nutzen der Daten für die Evaluationszwecke beeinträchtigt wird. Zur Anonymisierung solcher Daten existieren in der Literatur verschiedene Ansätze wie beispielsweise das Entfernen jeglicher identifizierender Merkmale oder das gänzliche Ausschließen von Suchanfragen, die persönliche Daten enthalten können.

Trotz aller hier genannten Nachteile und Einschränkungen werden Query Log Daten heute von vielen Forschungsgruppen zur Evaluation von Suchsystemen eingesetzt. Neben dem bereits erwähnten Hauptvorteil des deutlich geringeren manuellen Aufwands für die Generierung von Relevanzbeurteilungen und den damit verbundenen wesentlich größeren Datenmengen, liegt ein weiterer Vorteil dieser Methode in der Möglichkeit, kontext-sensitive Informationen miteinzubeziehen und damit auch kontext-sensitive Suchsysteme zu evaluieren. Dies ist in Evaluationsumgebungen, die auf dem Cranfield Prinzip (vgl.

Kapitel 3.2) beruhen, nicht möglich, da diese mit ausschließlich system-orientierten Relevanzbewertungen arbeiten, die Benutzerinteraktionen oder sonstige kontextuelle Informationen in keiner Weise berücksichtigen (White et al. 2010).
Struktur und Inhalt von Query Logs sind von mehreren Einflussfaktoren abhängig und variieren je nach dem zugrunde liegenden System, der Extraktionsmethode sowie den geplanten Einsatzszenarien. Gewisse grundlegende Informationen sind jedoch bei allen Query Logs vorhanden. Dazu zählen

(1) ein eindeutiger Bezeichner zur Identifikation von verschiedenen Benutzern.

(2) die konkrete Suchanfrage, wie sie vom Benutzer eingegeben wurde.

(3) eine Liste mit Suchergebnissen, die vom Suchsystem geliefert wurden inklusive ihres jeweiligen Rangs innerhalb der Liste sowie der Information, ob das Ergebnis vom Benutzer angeklickt wurde. Diese Information wird in der Literatur häufig auch als „Click-through Data“ bezeichnet.

(4) ein Zeitstempel mit genauen Datum und Uhrzeit der Suchanfrage.

In manchen Fällen wird zusätzlich zur Benutzerkennung auch eine sogenannte Session ID protokolliert. Eine Session (englisch für Sitzung) ist eine zeitlich begrenzte Abfolge von Suchanfragen und Systeminteraktionen eines Benutzers. Session Informationen werden dazu benutzt, um einzelne Benutzer/innen leichter identifizieren und zusammenhängende Suchanfragen besser abgrenzen zu können. Andere Systeme wiederum protokollieren nicht alle Suchergebnisse, sondern nur jene, die aktiv von den Benutzern/innen betrachtet (geklickt) wurden. Zur automatisierten Erfassung von Query Log Daten gibt es mehrere technische Ansätze. Der einfachste unter ihnen protokolliert im zentralen Suchserver einzig die Anfragen und Interaktionen der Anwender/innen. Um die suchenden Benutzer/innen aber eindeutig identifizieren zu können sind darüber hinausgehende Maßnahmen wie beispielsweise eine Benutzeranmeldung („User Login“) oder das Setzten von sogenannten Cookies notwendig. In manchen Fällen werden den Anwender/innen auch Zusatzprogramme („Search Toolbar“, „Browser Plug-In“) bereitgestellt, die auf dem Computer des Benutzers installiert werden.

Diese Zusatzprogramme kommen vor allem dann zum Einsatz, wenn neben den oben angeführten Daten noch tiefergehende Informationen gesammelt werden sollen. Es wurde gezeigt, dass Click-through Daten (vgl. Punkt 3) eine wichtige Grundlage zur Bildung von Relevanzbewertungen darstellen. Dieser Schritt kann durch Einbeziehen weiterer Kriterien noch verfeinert werden. Zu den zwei populärsten Verfeinerungskriterien zählen (1) die „Verweildauer" (engl. „Page Dwell Time"), die angibt, wie lange ein/e Benutzer/in ein geklicktes Suchergebnis betrachtet, bevor er/sie zur Ergebnisliste zurückkehrt oder die Suchseite schließt, sowie (2) die „Ausstiegsaktion" (engl. „Search Exit Action"), die Auskunft darüber gibt, auf welchem Wege der/die Benutzer/in die Suchseite verlassen hat (Browser geschlossen, auf andere Seite gewechselt, Session-Dauer abgelaufen, usw.).
Ungeachtet des großen Nutzens von Query Logs zur Generierung von Relevanzbewertungen müssen Störfaktoren, die beim Einsatz dieser Daten auftreten, unbedingt berücksichtigt und bereinigt werden. So konnte zum Beispiel gezeigt werden, dass Seiten, die weiter vorne im Suchergebnis gereiht werden, öfters angeklickt werden, als Seiten, die erst weiter hinten in der Ergebnisliste aufscheinen, auch wenn die hinteren Seiten eine höhere Relevanz in Bezug auf die Suchanfrage aufweisen (Croft et al. 2010).
Die absolute Klickhäufigkeit ist daher als Kennzahl für die Erstellung der Relevanzbewertung nicht ausreichend. In der Literatur werden mehrere Ansätze beschrieben, die diesem Umstand Rechnung tragen. Eine Möglichkeit ist beispielsweise die „Skip Above And Skip Next" Methode von Agichtein et al. (Eugene Agichtein et al. 2006), die auf der Annahme beruht, dass eine geklickte Seite P_c eine höhere Relevanz aufweist, als alle vor ihr gereihten sowie alle direkt nachfolgenden Seiten, auf die der/die Anwender/in nicht geklickt hat.

3.5 Personalisierte Suchsystemevaluation

Personalisierte Suchsysteme berücksichtigen neben der von den Anwendern/innen eingegebenen Suchanfrage noch weitere benutzerbezogene (kontextuelle) Parameter. Dies hat unweigerlich zur Folge, dass ein Suchsystem für ein und dieselbe Suchanfrage unterschiedliche Suchergebnisse in Abhängigkeit der suchenden Person liefert. Für eine detaillierte Beschreibung von kontext-sensitiven Suchalgorithmen und aktuellen Technologien sei an dieser Stelle auf Kapitel 0 hingewiesen.

Im Hinblick auf die Evaluation von personalisierten Systemen bestehen jedoch grundlegende Unterschiede im Vergleich zu „traditionellen" nicht personalisierten Systemen. Eine der fundamentalen Prinzipien der Cranfield Methode, wonach die Relevanz eines Dokuments für eine bestimmte Suchanfrage eine objektive und eindeutige Eigenschaft des Dokuments ist (vgl. Kapitel 3.2), kann im Kontext eines personalisierten Suchalgorithmus nicht mehr bestehen. Dementsprechend sind auch Korpora und Testdatensätze, die auf Grundlage der Cranfield Prinzipien erstellt wurden, nicht ausreichend, um personalisierte Systeme zu evaluieren. Vielmehr werden analog zu den personalisierten Suchergebnissen auch personalisierte Suchanfragen und personalisierte Relevanzbewertungen benötigt, die die jeweiligen Informationsbedürfnisse widerspiegeln und dem erweiterten, kontext-sensitiven Ansatz solcher Systeme Rechnung tragen (Harpale et al. 2010). Die Verfügbarkeit solcher Testdatensätze ist jedoch äußerst eingeschränkt. Viele bekannte und weit verbreitete Korpora, wie beispielsweise jene der TREC Reihe (vgl. Kapitel 3.3) basieren auf den Cranfield Grundlagen und verfügen dementsprechend weder über personalisierte Suchanfragen noch über personalisierte Relevanzbewertungen. Stattdessen werden sowohl Anfragen als auch Relevanzbewertungen von einer Gruppe von Personen erzeugt und zu einem einheitlichen Ergebnis aggregiert. Um die Relevanz eines Dokuments in Bezug auf eine Suchanfrage einer bestimmten Person beurteilen zu können, ist es jedoch notwendig, dass die suchende Person – und nur die suchende Person – die jeweiligen Dokumente hinsichtlich ihrer tatsächlichen Relevanz bewertet. Dies steht im Widerspruch zu allen oben erwähnten Testdatensätzen (Harpale et al. 2010).

Es ist daher wenig überraschend, dass viele Forschungsgruppen mangels der Verfügbarkeit von standardisierten Korpora für personalisierte Suchsysteme auf alternative Evaluationskonzepte wie beispielsweise proprietär gestaltete Testdatensätze und Benutzerexperimente zurückgreifen. Dabei rückte das Thema der „Folksonomy“ in den letzten Jahren zusehends in den Mittelpunkt.

3.6 Folksonomies zur Evaluation von personalisierten Suchsystemen

Der Begriff der „Folksonomy“ wurde zu Beginn der 2000er Jahre erstmals verwendet und beschreibt eine spezielle Form einer Taxonomie, die nicht von wenigen, speziell geschulten System- oder Domänen-Experten, sondern von vielen Endbenutzern ohne spezielles Vorwissen und ohne Vorgabe von besonderen Regeln oder Einschränkungen erstellt wird (Van Damme et al. 2001).

Als Bestandteil von sogenannten „sozialen Websites“ oder „Web 2.0 Anwendungen“ bieten Folksonomies den Website Benutzern die Möglichkeit, Inhalte individuell zu markieren (engl. Bookmark) und mit frei wählbaren Schlagwörtern (engl. Tags, Social Tags oder auch Social Annotations) zu versehen. Mit dieser einfachen aber wirkungsvollen Vorgehensweise können Anwender/innen Inhalte nicht nur organisieren und strukturieren sondern auch besonders interessante Dokumente kennzeichnen und vormerken. Genau diesen Umstand machen sich (Xu et al. 2008) zu nutze. Sie präsentierten bereits 2008 eine personalisierte Suchstrategie, die auf ihrer grundlegenden Annahme „Soziale Annotationen sind hochwertige Deskriptoren der in Websites enthaltenen Themen sowie gute Indikatoren für die Interessen der Website Besucher“ basierte. Im Zuge von mehreren durchgeführten Experimenten, die sie in ihrer Arbeit dokumentiert haben, konnten sie außerdem belegen, dass der Einsatz von Folksonomies zur personalisierten Suche zu einer deutlichen Verbesserung des Gesamtergebnisses beitragen kann.

Die Arbeit von Xu et al. ist an dieser Stelle aber vor allem deshalb von so hohem Interesse, weil sie eine eigene Evaluationsinfrastruktur für personalisierte Suchsysteme auf Basis von Folksonomy Daten entwickelt haben. Mit Hilfe dieser Entwicklung ist es ihnen gelungen eines der größten Probleme bei solchen Testdaten, nämlich das Generieren von personalisierten Suchanfragen und Relevanzbewertungen, zu lösen. Während bei Query Log-basierten Evaluationsmethoden (vgl. Kapitel 3.4) davon ausgegangen wird, dass von bestimmten Benutzerinteraktionen (zum Beispiel: „Anklicken eines Dokuments in den Suchergebnissen“) auf die Relevanz eines Dokuments in Bezug auf eine Suchanfrage für einen Anwender geschlossen werden kann, definieren Xu et al., dass „die Bookmarking und Tagging Aktionen eines Benutzers als dessen persönliche Relevanzbewertungen interpretiert werden können“.

Wenn also ein/e Benutzer/in ein Lesezeichen (engl. Bookmark) für die Website der Alpen-Adria-Universität Klagenfurt (http://www.uni-klu.ac.at) erstellt und dafür das Schlagwort (engl. Social Tag) „Computerlinguistik“ vergibt, wird daraus geschlossen, das der/die Benutzer/in im Umkehrschluss die Website der Alpen-Adria-Universität Klagenfurt in der Ergebnisliste finden möchte, wenn er nach „Computerlinguistik“ sucht. Umgekehrt räumen die Autoren aber ein, dass nicht behauptet werden kann, dass eine Website für den/die Benutzer/in generell nicht relevant wäre, nur weil er keine Beschlagwortung für sie erstellt hat. Dies ist jedoch ein generelles Problem bei der Erstellung von Testdaten in diesem Umfeld und trifft demnach auch in gleicher Weise auf Korpora zu, die keine Folksonomy-basierten Daten zugrunde legen.

Zur Untermauerung ihrer grundlegenden Annahme („Aus Social Tags können persönliche Relevanzbewertungen für die Evaluation von Suchsystemen extrahiert werden“) führen (Xu et al. 2008) drei wesentliche Argumente an:

(1) Die häufigste Repräsentationsform für Suchanfragen bei aktuellen Informationssuchsystemen ist jene der Stichwortanfrage (engl. Keyword Query). Da Social Tags in Folksonomy Daten auch Stichwörter in Bezug auf das beschlagwortete Dokument darstellen, können diese ebenfalls als Suchanfragen (engl. Query) interpretiert werden.

(2) Ein Dokument oder eine Website kann mehrere verschiedene Themen beinhalten. Dementsprechend können sich unterschiedliche Benutzer/innen für unterschiedliche Themen eines Dokuments oder einer Website interessieren. Diese Benutzer/innen werden mit sehr großer Wahrscheinlichkeit auch unterschiedliche Tags für das gleiche Dokument beziehungsweise die gleiche Website vergeben. Wenn diese Tags als Suchanfragen interpretiert werden, bedeutet dies, dass unterschiedliche Benutzer/innen ein Dokument oder eine Website für unterschiedliche Suchanfragen als relevant betrachten.

(3) Personen verwenden jene Begriffe bei der Beschlagwortung von Inhalten, die ihrem persönlichen Sprachgebrauch am ehesten entsprechen. Dies führt dazu, dass unterschiedliche Benutzer/innen unterschiedliche Social Tags für die gleichen Inhalte verwenden und hat dadurch den Vorteil, dass Evaluationsergebnisse nicht nur vordefinierte Begrifflichkeiten verzerrt werden.

Die eigentlichen Daten für das oben beschriebene Evaluationsframework haben Xu et al. aus zwei unterschiedlichen Quellen bezogen. Zum einem wurden vom Online Bookmarking Dienst „Del.icio.us“[6] Inhalte aus über 90.000 Websites mit mehr als 65.000 unterschiedlichen Schlagwörtern (Tags) von 9.800 Benutzern gesammelt. Zum anderen wurden über 79.000 Websites mit fast 48.000 eindeutigen Schlagwörtern von mehr als 5.000 Anwendern aus dem Dogear System[7], ein Forschungsprojekt der Watson Gruppe von IBM Research[8], extrahiert. Während der erste Datensatz vor allem allgemeine Inhalte und Informationsbedürfnisse von Webanwendern repräsentiert, spiegelt der zweite Datensatz explizit die Verhaltensmuster von Anwendern in Unternehmen wider (Millen et al. 2006). Für die konkrete Durchführung der Evaluationsstudien haben Xu et al. aus beiden der oben angeführten Datensammlungen je drei Testdatensätze erstellt. Die Zuordnung der Quelldaten in die unterschiedlichen Testdatensätze erfolgte dabei anhand der Anzahl von Bookmarks pro Benutzer/in. So enthält beispielsweise der Testdatensatz „DEL.80-100“

6 https://delicious.com/

7 http://researcher.watson.ibm.com/researcher/view_project.php?id=2244

8 http://www.watson.ibm.com

alle Datensätze und Schlagwörter von 100 zufällig gewählten Benutzern die jeweils zwischen 80 und 100 Lesezeichen (Bookmarks) zugeordnet haben. Die nachfolgende Tabelle zeigt eine detaillierte Auflistung aller generierten Testdatensätze mit einer detaillierten Beschreibung ihrer wesentlichen Parameter Anzahl der Benutzer („Num. Users“), maximale Anzahl der vergebenen Schlagwörter („Max. Tags“), minimale Anzahl der vergebenen Schlagwörter („Min. Tags“), durchschnittliche Anzahl vergebener Schlagwörter („Avg. Tags“), maximale Anzahl der beschlagworteten Seiten („Max. Pages“), minimale Anzahl der beschlagworteten Seiten („Min. Pages“) und durchschnittliche Anzahl der beschlagworteten Seiten („Avg. Pages“).

Data Set	Num. Users	Max. Tags	Min. Tags	Avg. Tags	Max. Pages	Min. Pages	Avg. Pages
Delicious	9813	2055	1	56.04	1790	1	40.35
Dogear	5192	2288	1	47.43	4578	1	46.78
DEL.gt500	31	1133	74	464.42	1790	506	727.55
DEL.80-100	100	456	2	107.51	100	80	88.43
DEL.5-10	100	64	1	18.53	10	5	7.44
DOG.gt500	92	2147	42	543.87	4578	500	999.04
DOG.80-100	85	295	9	126.96	100	80	89.32
DOG.5-10	100	41	2	16.11	10	5	6.99

Tabelle 6: Auflistung aller Testdatensätze und deren Eigenschaften; Quelle: (Xu et al. 2008)

Bevor die Datensätze endgültig für die Evaluationsdurchführungen verwendet werden konnten, mussten von den Autoren noch zwei Bereinigungsschritte durchgeführt werden. Zum ersten wurden Schlagwörter ohne inhaltliche Aussagekraft wie beispielsweise „toread“ oder „imported_IE_Favorites“ mittels eines manuell erstellen Regelwerks entfernt. Zum zweiten wurden Schlagwörter, die aus mehreren von Benutzern verbundenen Begriffen, wie zum Beispiel: „javaprogramming“, entstanden sind, unter Einsatz eines Wörterbuchs wieder aufgetrennt.

Die in (Xu et al. 2008) beschriebene Methode zur Gewinnung von Testdatensätzen für die Evaluation von personalisierten Suchsystemen ermöglicht in der Tat eine signifikante Reduktion des Aufwands bei der Erstellung von Suchanfragen und Relevanzbewertungen und bietet damit völlig neue Möglichkeiten. Bis dahin konnten personalisierte Suchsysteme nur mittels individueller Benutzerstudien oder Query Log-basierter Testkorpora

evaluiert werden. Wobei erstere Methode den Nachteil hat, dass eine ausreichend große Menge an teilnehmenden Personen benötigt wird. Dies verursacht jedoch hohe Kosten und birgt darüber hinaus das Risiko, dass Testergebnisse verzerrt werden können, da sich die Teilnehmer/innen darüber bewusst sind, dass sie getestet werden, und sich dadurch anders verhalten.

Ein weiterer Nachteil besteht darin, dass die Testergebnisse nicht reproduziert und nur eingeschränkt verglichen werden können. Das Problem bei der Query Log Methode wiederum ist, dass dafür große Mengen an Log Daten benötigt werden, die aber auf Grund von eigentums- und datenschutzrechtlichen Gründen nur für die wenigsten Forscher/innen zugänglich sind (vgl. Kapitel 3.4). Die oben beschriebene Methode von Xu et al. kann für alle hier angeführten Probleme eine Lösung bieten. Leider wurden aber bisher weder die generierten Testkorpora noch die zugrundeliegenden Datensammlungen („Del.icio.us“, „Dogear“) veröffentlicht. Der Autor dieser Dissertation hat sowohl bei den Autoren von (Xu et al. 2008) als auch bei den Hauptverantwortlichen Entwicklern des Dogear Systems (Millen et al. 2006) angefragt. Eine Veröffentlichung oder zumindest spezifische Bereitstellung der Daten ist aber leider aus unterschiedlichen Gründen nicht möglich und auch für die Zukunft nicht geplant.

3.7 Das CiteData Korpus

Wie eingangs bereits erwähnt, ist eine standardisierte, vergleichbare und reproduzierbare Evaluationsinfrastruktur eine wichtige Grundlage für die langfristig erfolgreiche Weiterentwicklung von Suchsystemen. Ein Ansatz, der diesem Problem Abhilfe schaffen will, ist das „CiteData“ Korpus von Harpale et al. (Harpale et al. 2010). Es enthält eine umfangreiche Dokumentsammlung von akademischen Papieren, personalisierte Suchanfragen und Relevanzbewertungen sowie zusätzliche Metainformationen in Form von Doku-

mentverlinkungen, Kategoriebezeichnungen und sogenannten „Social Tags“ (vgl. Kapitel 3.6). Das Korpus ist zudem frei verfügbar und kann von der Website[9] der Autoren heruntergeladen werden.

Es muss jedoch an dieser Stelle festgehalten werden, dass bis zum Zeitpunkt des Abschlusses dieser Arbeit noch nicht alle Artefakte des Datensatzes verfügbar waren, da sich das Projekt noch in der Entwicklungsphase befindet. Laut Auskunft der Autoren werden die noch fehlenden Informationen aber ebenfalls zum Download bereitgestellt, sobald deren Entwicklung abgeschlossen ist. Ein genauer Veröffentlichungszeitpunkt ist jedoch noch nicht bekannt. Im Folgenden wird das CiteData Korpus von Harpele et al. sowie dessen Nutzen bei der Validierung von personalisierten Suchsystemen detailliert beschrieben.

Um einen der bestehenden standardisierten Testdatensätze – wie beispielsweise TREC – mit allen für personalisierte Suchsystemevaluationen benötigten Daten (vor allem personalisierte Relevanzbewertungen) anzureichern, wären sehr umfangreiche Benutzerstudien nötig. Daher haben sich Harpele et al. dazu entschlossen, ein neues Korpus auf Basis der Social Bookmarking Website „CiteULike“[10] zu erstellen. Diese Website erlaubt es ihren Benutzern, akademische Artikel, die für sie von besonderem Interesse sind, zu merken (engl. bookmark). Zusätzlich können die Benutzer/innen für jeden gemerkten Artikel beliebig viele frei definierbare Schlagwörter (engl. Tags, Social Tags oder auch Social Annotations) vergeben.

9 http://nyc.lti.cs.cmu.edu/datasets/citedata/

10 http://www.citeulike.org/

Abbildung 4: Exemplarischer Bookmark mit Metainformationen und persönlichen Social Tags in CiteULike

Abbildung 4 zeigt einen Screenshot eines Bookmarks im CiteULike System. Dieser Bookmark wurde mit dem persönlichen Benutzerkonto des Autors im System erstellt und bezieht sich auf die Publikation mit dem Titel „Exploiting Social Tagging Profiles to Personalize Web Search“. Als Metainformationen werden zusätzlich die Liste der Autoren („by: David Vallet, [...]“), die Liste der Editoren („edited by: Troels Andreasen, [...]“), das Veröffentlichungsmedium („In Flexible Query Answering Systems, Vol [...]“) sowie weitere Dokumentdetails („doi“, „citeulike Key“) angezeigt. Die Kurzfassung (Abstract) des Dokuments wird ebenfalls von CiteULike bereit gestellt. Diese ist jedoch nicht bei allen Artikeln verfügbar. In der letzten Zeile des Screenshots sind unter der Überschrift „My tags for this article“ außerdem die Schlagwörter (Social Tags) „judgement“, „personalized“, „relevance“, „search“, „social“ und „tagging“ ersichtlich, die der Autor für diesen Artikel beziehungsweise diesen Bookmark vergeben hat.

Harpale et al. haben sich für diese Website als primäre Datenquelle entschieden, da die akademischen Artikel natürlichsprachliche textuelle Inhalte aus verschiedenen Themengebieten darstellen, die Verweise und Zitate zwischen den Artikeln als Dokumentverlinkungen interpretiert werden können und reichhaltige Metainformationen, zum Beispiel in Form von Social Tags, existieren.

Die darüber hinaus benötigten personalisierten Suchanfragen und Relevanzbeurteilungen können jedoch nicht von CiteULike extrahiert werden. Diese Informationen werden daher

einerseits von der digitalen Publikationsdatenbank „CiteSeer“[11] und andererseits von manuellen Eingaben freiwilliger Studienteilnehmer/innen gewonnen. Da CiteSeer aber hauptsächlich Dokumente aus den Bereichen Informatik und Software Entwicklung enthält, und die Ressourcen für die manuellen Such- und Relevanzbewertungserstellungen eingeschränkt waren, wurde aus der gesamten, in CiteULike verfügbaren Dokumentenmenge von ungefähr 800.000 Stück nur eine Teilmenge von 81.400 Informatik-affinen Publikationen für die weitere Verwendung im Korpus selektiert. Die Autoren argumentieren diese Einschränkung außerdem damit, dass dadurch die Qualität der personalisierten Relevanzbewertungen sehr hoch gehalten werden kann, da es sich bei den freiwilligen Studienteilnehmer/inne/n ebenfalls um Forschungsmitarbeiter/innen und Student/inn/en aus dem Gebiet der Informatik handelt.

Das CiteSeer System bietet eine umfangreiche Sammlung von akademischen Artikeln aus dem Bereich der Informatik und ist innerhalb der Forschungsgemeinde weithin als seriöse und qualitativ hochwertige Datenquelle anerkannt. Die CiteSeer Online-Bibliothek ist frei zugänglich und enthält neben den eigentlichen Artikeln noch umfangreiche Metadaten wie beispielsweise Artikelkurzfassungen (engl. Abstract) und Informationen über Autoren, Erscheinungsjahr, Erscheinungsmedium, uvm. Im Gegensatz zum CiteULike System, können die Artikelinformationen in CiteSeer nicht öffentlich beliebig bearbeitet werden. Folglich sind die Daten in diesem System vollständiger, korrekter und vor allem frei von Spam oder sonstigen Verunreinigungen und eignen sich daher besser als Extraktionsquelle für die gewünschten Testdaten. Darüber hinaus haben die Autoren in (Harpale et al. 2010) noch alle Referenzen sämtlicher Publikationen extrahiert und daraus einen Dokumentverlinkungsgraphen generiert. Diese Informationen sollten vor allem für Suchsysteme verwendet werden, die auf der Analyse von Interdokumentverknüpfungen - wie zum Beispiel der Personalized PageRank Algorithmus (Kamvar et al. 2003) – basieren.

Wie bereits erwähnt, bietet CiteULike die Möglichkeit, Lesezeichen mit frei wählbaren textuellen Schlagwörtern (engl. „Tags“) anzureichern. Dabei protokolliert das System für jedes vergebene Tag *t* den Artikel *a*, auf den sich *t* bezieht, sowie den Benutzer *u*, der *t*

[11] http://csxstatic.ist.psu.edu/

vergeben hat und den Zeitpunkt *s* an dem *u* den Tag *t* zugeordnet hat. Diese Daten stehen als 4-dimensionale Tupel < a, s, u, t > öffentlich auf der CiteULike Dataset Website[12] zum Download zur Verfügung.

Diese Daten mussten jedoch einer Reihe von Bereinigungsschritten unterzogen werden, ehe sie im CiteData Korpus eingesetzt werden konnten. Wie in (Harpale et al. 2010) erläutert, beinhaltet der CiteULike Datensatz zahlreiche Schlagwörter, die von externen Programmen (sogenannte „Robots“[13]) automatisiert vergeben werden, die aber keinerlei inhaltliche Aussagekraft besitzen. Weitere Verunreinigungen – im Sinne von Schlagwörtern ohne inhaltliche Aussagekraft – entstehen in den CiteULike Daten vor allem durch unerwünschte Werbetätigkeiten (engl. Spam). Daher wurden die originalen Daten gefiltert und nur jene Datensätze berücksichtigt, die von „echten Benutzern“ vergeben worden sind. Zur Identifikation von relevanten Datensätzen wurden spezielle Heuristiken verwendet. Die wichtigsten Kriterien waren dabei unter anderem

(1) die Anzahl der echten Benutzer/innen, die einen Artikel als Lesezeichen markiert haben sowie

(2) die Anzahl der Lesezeichen, die ein/e Benutzer/in im System insgesamt eingetragen hat

Als „echte Benutzer/innen“ wurden beispielsweise nur jene Anwender/innen gewertet, die nicht weniger als vier und nicht mehr als 500 Artikel zur ihren Lesezeichen hinzugefügt hatten. Weiters wurden alle Artikel entfernt, die von weniger als vier „echten Benutzern“ mit einem Lesezeichen markiert wurden. Für eine detaillierte Beschreibung der Vorgehensweise verweisen die Autoren auf (Jin & Si 2004), wonach solche Bereinigungsstrategien im Bereich des „Collaborative Filterings“ häufig zur Erstellung von Testdatensätzen herangezogen werden.

Im Anschluss an die Datenbereinigung wurden im nächsten Schritt für alle verbliebenen Artikel Klassifikationszuordnungen erstellt. Bei ungefähr 6.000 Artikeln konnten diese

[12] http://www.citeulike.org/faq/data.adp

[13] http://www.robotstxt.org/

Informationen laut Auskunft der Autoren direkt von CiteSeer mittels der frei verfügbaren und öffentlich zugänglichen „CiteSeer Classification Hierarchy“ bezogen werden. Für alle anderen Artikel, für die keine explizite Kategorieinformation in CiteSeer verfügbar war, wurde eine automatisierte Kategorisierung mit Hilfe des polynomialen „SVM^{light}“[14] Klassifizierers von Thorsten Joachim[15] durchgeführt. Als Trainingsdaten wurden hierfür die existierenden Kategorieinformationen aus CiteSeer herangezogen.
Abbildung 5 zeigt eine Liste der zwölf Kategorien, die im CiteData Korpus verwendet werden, sowie die Verteilung der Kategoriezuordnungen als relative Häufigkeit der zugeordneten Artikel.

ID	Category Title
1	Computer Programming
2	Machine Learning and AI
3	Networking and Security
4	Computer Architecture
5	Agents and Applications
6	Computer Theory
7	Databases
8	Human Computer Interaction
9	Digital Libraries
10	Web and Information
11	Natural Language Processing
12	Other research areas in Computer Science

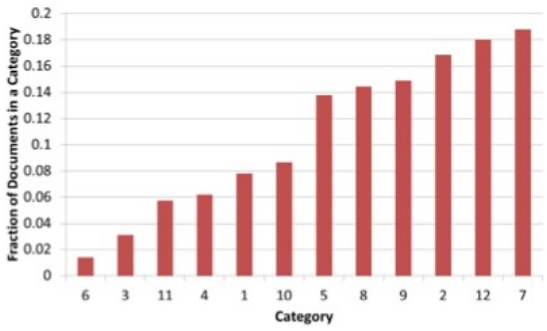

Abbildung 5: Liste der in CiteData vorhandenen Kategorien (links) sowie die relative Verteilung aller Artikel auf die Kategorien (rechts); Quelle: (Xu et al. 2008)

Zusätzlich zu den automatisch extrahierten Daten aus CiteULike und CiteSeer wurden Experten eingeladen, um individuelle Suchaufgaben und Relevanzberwertungen zu definieren und das CiteData Korpus entsprechend manuell zu annotieren. Die Teilnehmer/innen wurden hierfür aus dem Kreise der wissenschaftlichen Mitarbeiter/innen und Doktoratsstudent/inn/en des Informatikbereichs ausgewählt. Bei der Selektion der Teilnehmer

[14] SVM ... Support Vector Machine

[15] http://svmlight.joachims.org

wurde einerseits darauf geachtet, dass sie über das notwendige Fachwissen verfügen, um den Inhalt der zu annotierenden Publikationen verstehen zu können. Andererseits wurde auch darauf geachtet, dass im Testkorpus ausreichend viele Artikel zum Fachgebiet des jeweiligen Experten vorhanden waren.

Um den Anforderungen aus dem täglichen Leben am nächsten zu kommen, wurden zu Beginn der Studie alle Teilnehmer/innen aufgefordert, sich eine übergeordnete Rechercheaufgabe zu überlegen und diese in Form eines „Task Statements" niederzuschreiben. Ausgehend von den selbstdefinierten Task Statements erstellten die Teilnehmer/innen jeweils vier bis sechs Suchanfragen (engl. Query), mit deren Hilfe das System durchsucht wurde. Anschließend wurden die Dokumente in den Ergebnislisten von den Teilnehmer/inne/n betreffend ihrer Relevanz in Bezug die Suchanfrage annotiert.

Harpale et al. dokumentieren in ihrer Arbeit aber nicht nur den Inhalt und die Entstehung des CiteData Evaluationskorpus sondern präsentieren darüber hinaus auch einen Test von mehreren personalisierten Suchstrategien: Im Zuge dessen wurden vier personalisierte Suchstrategien (1) „Matching User's Topical Interest to Document Categories", (2) „Personalized Page Rank", (3) „Personalized Collaborative Filtering" und (4) „Meta Personalized Search" und zwei nicht-personalisierte Strategien (a) „Indri Retrieval" und (b) „General Page Rank" auf Basis des CiteData Korpus evaluiert und die Testergebnisse mittels dem „Mean Average Precision" (MAP) Maß (Details zu MAP siehe Kapitel 3.10) miteinander verglichen. Der Vergleich ergab, dass alle untersuchten personalisierten Methoden bessere Ergebnisse erzielen, als die nicht personalisierten, und dass die Kombination aus mehreren personalisierten Strategien (entspricht Strategie 4 „Meta Personalized Search") die besten Testergebnisse liefert.

UserID	network03
Task	Information Network Security
Task Statement	Access control is the process in which a request to a data resource or service is mediated to determine whether the access should be granted or denied....
Query1	role based access control
Query2	workflow access control
Query3	authorization delegation
Query4	distributed access control
Query5	XML access control

Abbildung 6: Exemplarisches Task Statement für die Suchaufgabe "Information Network Security / Access Control" mit seinen zugehörigen Suchanfragen (Query1 ... Query5); Quelle: (Xu et al. 2008)

Die Verfügbarkeit eines standardisierten, frei zugänglichen Testkorpus für die Evaluation von personalisierten Suchsystemen stellt einen wichtigen Bestandteil für das Forschungsgebiet der Informationssuche dar. Harpale et al. beschreiben in ihrer Publikation (Harpale et al. 2010) nicht nur, wie solche Datensätze durch die Kombination von traditionellen Vorgehensweisen (manuell erstellte Suchanfragen und Relevanzbewertungen) und Folksonomy-basierten Methoden (Extraktion von Social Tags und anderen Metadaten aus den Web 2.0 Applikationen CiteSeer und CiteULike) generiert werden können, sondern stellen diese Daten auch in Form des CiteData Korpus anderen Forschern zur Verfügung. Darüber hinaus zeigen sie in ihrer Arbeit auch konkret, wie dieses Korpus zur Evaluation unterschiedlicher personalisierter und nicht-personalisierter Suchstrategien eingesetzt werden kann.

3.8 Metriken für die Effektivitätsbewertung

Zur Effektivitätsbeurteilung von Suchsystemen gibt es zahlreiche Bewertungsmetriken. Die beiden am häufigsten verwendeten Maße dabei sind die Genauigkeit (engl.: Precision) und die Trefferquote (auch Vollständigkeit; engl.: Recall). In diesem Kapitel wird die Berechnung, Bedeutung und der Einsatz der beiden Kennzahlen folgend der Beschreibung in (Croft et al. 2010) erläutert.

Precision und Recall wurden erstmals im Zuge der Cranfield Experimente (vgl. Kapitel 3.2) beschrieben beziehungsweise verwendet und dienen heute als Grundlage für viele Evaluationskennzahlen im Bereich der Informationssuche. Vereinfacht gesprochen gibt Precision an, wie viele Dokumente im Suchergebnis wirklich relevant sind, während mit Recall ausgedrückt wird, wie viele der insgesamt relevanten Dokumente vom System gefunden und zurück geliefert wurden. Bei der Berechnung der beiden Maße wird davon ausgegangen, dass für jede Suchanfrage, die an ein System übermittelt wird, aus der Gesamtmenge aller Dokumente eine Teilmenge an relevanten Dokumenten als Suchergebnis zurückgeliefert wird.

Sofern die Relevanz von Dokumenten für eine Suchanfrage als binäre Eigenschaft (Dokument ist entweder relevant oder nicht relevant) definiert wird, können die möglichen Ergebnismengen des untersuchten Systems wie in Tabelle 7 dargestellt zusammengefasst werden. Dabei repräsentiert die Teilmenge *A* jene Dokumente, die als tatsächlich relevant für eine Suchanfrage *q* definiert sind („Relevant“). Die Menge *B* hingegen beinhaltet alle Dokumente, die vom System als relevant erachtet und somit als Teil des Suchergebnisses für eine Anfrage *q* zurück geliefert werden („Im Suchergebnis“). Im Gegensatz dazu entspricht *A'* der Menge der Dokumente, die für *q* nicht relevant sind („Nicht relevant“) und *B'* der Menge der Dokumente, die vom System als nicht relevant identifiziert und somit nicht als Teil des Suchergebnisses zurück geliefert werden („Nicht im Suchergebnis“). Dementsprechend definiert der Ausdruck $A \wedge B$ die Menge aller Dokumente, die als relevant definiert (*A*) und vom System auch als solche erkannt (*B*) werden. Analog dazu gibt $A' \wedge B'$ die Menge der Dokumente an, die in der Gesamtmenge als nicht relevant für *q* gelten (*A'*) und vom System auch nicht in das Suchergebnis aufgenommen werden (*B'*).

	Relevant	Nicht relevant
Im Suchergebnis	$A \wedge B$	$A' \wedge B$
Nicht im Suchergebnis	$A \wedge B'$	A' ∧ B'

Tabelle 7: Einteilung der möglichen Ergebnismengen von Suchsystemen bei binärer Relevanzbeurteilung; Quelle: (Croft et al. 2010)

Die Menge $A \wedge B'$ wird oft als „False Negative" bezeichnet, da das System Dokumente als nicht relevant markiert und folglich nicht in das Suchergebnis aufgenommen hat (B'), obwohl sie für die Suchanfrage q als relevant gelten (A). Umgekehrt umfasst $A' \wedge B$ jene Dokumente, die zwar in Bezug auf q nicht relevant sind (A'), aber vom System dennoch in die Ergebnismenge aufgenommen werden (B). Man nennt diese Fälle häufig auch „False Positive". Ausgehend von der in Tabelle 7 beschriebenen Logik, können die beiden Kennzahlen Precision und Recall wie folgt definiert werden:

Die Kennzahl Precision p (Genauigkeit) entspricht dem Anteil der relevanten Dokumente im Suchergebnis für eine Suchanfrage q. Umso höher der Wert für p, desto höher ist der Anteil der relevanten Dokumente im Suchergebnis im Verhältnis zu allen gelieferten Treffern, und desto höher ist die Genauigkeit des Suchsystems. Ein p Wert von 1 würde demnach bedeuten, dass alle vom System zurück gelieferten Dokumente korrekt (d. h. relevant) sind. Die vollständige Formel zur Beschreibung von p ist in Gleichung 2 dargestellt.

$$p = \frac{|A \wedge B|}{|B|}$$

Gleichung 2: Precision p als Verhältnis der Menge aller relevanten und zurück gelieferten Dokumente zur Menge aller zurück gelieferten Dokumente

Die Kennzahl Recall r (Trefferquote, Vollständigkeit) entspricht dem Anteil der relevanten Dokumente, die vom System insgesamt gefunden und zurück geliefert wurden an der Menge aller in der Datensammlung existierender relevanten Dokumente. Umso höher der

Wert für *r*, desto größer ist die Anzahl der relevanten Dokumente, die das System im gesamten Korpus identifiziert hat. Ein *r* Wert von 1 würde also bedeuten, dass das System alle korrekten (d. h. relevanten) Dokumente in der Datensammlung gefunden und diese zurück geliefert hat. Die vollständige Formel zur Berechnung von *r* kann der Gleichung 3 entnommen werden.

$$r = \frac{|\, A \,\wedge\, B \,|}{|\, A \,|}$$

Gleichung 3: Recall r als Verhältnis der Menge aller relevanten und zurückgelieferten Dokumente zur Menge aller insgesamt existierenden relevanten Dokumente

Die Berechnung der beiden oben angeführten Kennzahlen soll anhand des folgenden Beispiels noch verdeutlicht werden: Für eine Suchanfrage q_1 existieren in der Dokumentensammlung *M* insgesamt 19 als relevant definierte Dokumente. Das evaluierte System liefert ein Suchergebnis s_1, welches insgesamt 15 Dokumente enthält. Unter den 15 Dokumenten befinden sich 11 der insgesamt 19 als relevant definierten Dokumente. Der Wert für Precision *p* würde sich in diesem Fall auf 0,734 (entspricht 11 / 15) und der Wert für Recall *r* auf 0,579 (entspricht 11 / 19) belaufen.

Auf dieser grundlegenden Logik lassen sich auch noch viele weitere Kennzahlen, wie beispielsweise der Anteil aller nicht relevanten zurück gelieferten Suchergebnisse im Verhältnis zu allen existierenden nicht relevanten Dokumenten (engl. „Fallout“), definieren. Obwohl dieses grundlegende Konzept gewisse Schwachstellen beziehungsweise Einschränkungen aufweist (Powers 2011), haben sich in der wissenschaftlichen Praxis Precision und Recall sowie eine Reihe von darauf aufbauenden Kennzahlen und die Kombination aus diesen als de facto Standard für die Bewertung von Effektivitätskriterien bei Informationssuchsystemen herausgebildet.

3.9 Das F-Maß

Innerhalb der wissenschaftlichen Community gibt es ein breites Einverständnis darüber, dass Informationssuchsysteme im Allgemeinen als Resultat auf eine Suchanfrage so viele relevante Elemente wie möglich und dabei so wenig nicht relevante Elemente wie möglich zurück liefern sollten. Vereinfacht gesprochen entspricht die erste Vorgabe dabei dem Konzept von Recall und die zweite dem der Precision (vgl. Kapitel 3.8). Diese beiden Bedingungen beeinflussen sich bei der Realisierung von Suchsystemen gegenseitig, was zu ungünstigen Wechselwirkungen führt: Möchte man die Genauigkeit eines Systems erhöhen, indem man die Anzahl der nicht relevanten Suchergebnisse reduziert (entspricht: Anhebung des Precision Wertes), erhöht sich in vielen Fällen aber auch die Menge der False Negative Dokumente, also jener Dokumente, die zwar relevant für eine Suchanfrage sind, aber vom System nicht (mehr) als solche identifiziert werden. Dies wiederum führt zu einer Verringerung des Recall Wertes. Umgekehrt führt eine Anhebung des Recall Wertes oft zu einer verringerten Genauigkeit (Precision). Man spricht in diesem Zusammenhang auch von einer verringerten Selektivität des Systems. In Anbetracht dieser Situation ergibt sich nun die Frage, wie die Effektivität mehrerer unterschiedlicher Suchsysteme auf Basis der beiden genannten Kennzahlen verglichen werden kann beziehungsweise unter welchen Bedingungen eine Aussage darüber getroffen werden kann, welches System das „bessere“ ist. Ein möglicher Ansatz zur Beantwortung dieser Frage ist der Einsatz eines sogenannten Recall-Precision Graphs.

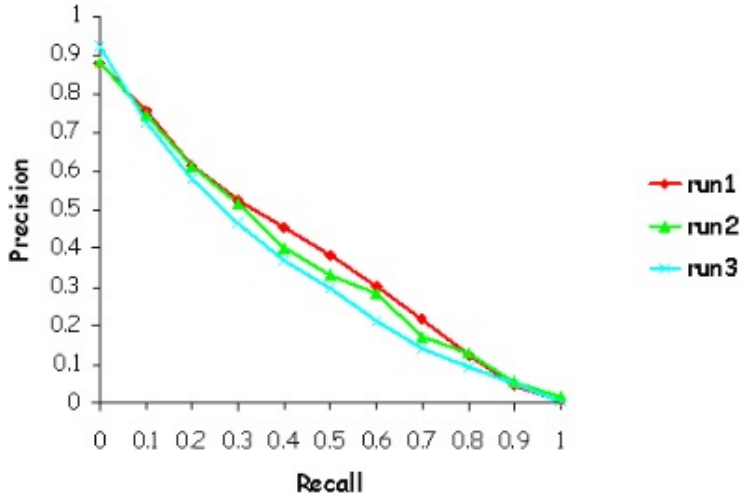

Abbildung 7: Exemplarischer Recall-Precision Graph mit drei unterschiedlichen Ergebniskurven; Quelle (Voorhees 1999)

Dabei wird die Precision *p* eines Systems bei jedem Recall Wert *r* ($0 \leq r \leq 1$) in einem bestimmten Intervall ermittelt. Zwei unterschiedliche Systeme können dann auf Basis dieses Graphs verglichen werden, wobei gilt, dass ein System S_1 dann „besser" (d.h. effektiver) ist als ein System S_2, wenn der *p* Wert von S_1 an jeder Stelle von *r* besser ist, als jener von S_2. Ist dies nicht der Fall, werden die einzelnen *p* Werte der beiden zu vergleichenden Systeme in Abhängigkeit von *r* ermittelt und anschließend der Durchschnittswert gebildet. Dieser Durchschnittswert kann anhand unterschiedlicher mathematischer Verfahren berechnet werden und stellt sodann die Basis für den Vergleich zwischen den beiden Systemen dar. (Raghavan et al. 1989)

Ein in der wissenschaftlichen Forschung sehr weit verbreitetes Maß zur Effektivitätsbewertung von Suchsystemen ist das sogenannte F-Maß. Wie in (Croft et al. 2010) erläutert, erlaubt dieses Maß, die Effektivität eines Systems als Kombination von Precision und Recall in einer einzigen Kennzahl auszudrücken. Aus mathematischer Sicht entspricht das F-Maß dem harmonischen Mittel gemäß folgender Gleichung:

$$F = \frac{2 * R * P}{R + P} \quad R \dots Recall, P \dots Precision$$

Gleichung 4: Berechnung des F-Maßes als harmonisches Mittel von Precision und Recall

Die Verwendung des harmonischen Mittels hat gegenüber dem arithmetischen Mittel den Vorteil, dass das Ergebnis robuster hinsichtlich numerischer Ausreißer beziehungsweise sehr großer Zahlenwerte im Allgemeinen ist. Der Einsatz des arithmetischen Mittels würde Systeme bevorteilen, die sehr unterschiedliche Werte für Precision und Recall erzeugen. So würde ein System mit einem Precision Wert von 1 aber einem Recall Wert von beinahe 0 bei Verwendung eines arithmetischen Mittels noch immer einen Wert von circa 0,5 erreichen. Bei der Berechnung des F-Maßes mittels des harmonischen Mittels würde das gleiche System hingegen nur noch einen Wert von deutlich kleiner 0,5 erreichen, was die Effektivität des Systems angemessener widerspiegelt.
Informationssuchsysteme werden heute in sehr unterschiedlichen Anwendungsszenarien eingesetzt. Dementsprechend unterschiedlich sind auch die Anforderungen ihrer Benutzer/innen hinsichtlich der Genauigkeit (Precision) und der Vollständigkeit (Recall) der vom System gelieferten Suchergebnisse. So sehen sich beispielsweise Internetsuchmaschinen mit äußerst großen Datensammlungen konfrontiert, während die Benutzer/innen meist nur an einer verhältnismäßig kleinen Menge an Suchergebnissen interessiert sind. Daher wird bei solchen Systemen der Schwerpunkt auf einen möglichst hohen Precision Wert gelegt. Umgekehrt kann es zum Beispiel bei medizinischen Informationssystemen von essentieller Bedeutung sein, alle relevanten Elemente zu finden. In diesen Fällen liegt der Fokus bei der Systemevaluation beziehungsweise –entwicklung klar auf der Recall Seite. Um diesem Umstand Rechnung zu tragen, wird daher die Berechnung des F-Maßes häufig um eine Gewichtung ergänzt. Man spricht dann von einem gewichteten harmonischen Mittel. Gleichung 5 zeigt die Formel zu Berechnung des gewichteten F-Maßes:

$$F = \frac{R * P}{\alpha * R + (1 - \alpha) * P} \quad R \dots Recall, P \dots Precision, \alpha \dots Gewicht$$

Gleichung 5: Berechnung des F-Maßes mit Hilfe des gewichteten harmonischen Mittels von Precision und Recall

In der wissenschaftlichen Praxis wir das Gewicht α häufig in der Form

$$\alpha = \frac{1}{\beta^2 + 1}$$

repräsentiert, wodurch sich für die Berechnung des F-Maßes in Abhängigkeit vom Gewicht β folgende Formel ergibt:

$$F_\beta = \frac{(\beta^2 + 1) * R * P}{R + \beta^2 * P} \quad R \dots Recall, P \dots Precision, \beta \dots Gewicht$$

Gleichung 6: Berechnung des F-Maßes in Abhängigkeit des Gewichts β

Durch die Erweiterung der Berechnungsformel für das F-Maß um das Gewicht β, lässt sich nun die relative Wichtigkeit der beiden Kennzahlen (Precision und Recall) zueinander parametrisieren. Wie aus obiger Gleichung ersichtlich, wird die Wichtigkeit von Recall umso höher, desto größer der Wert des Gewichts β gewählt wird, während gleichzeitig die Bedeutung von Precision abnimmt. Das F-Maß mit einem Gewicht von 1 wird häufig auch als F_1 bezeichnet und bedeutet, dass die relative Wichtigkeit von Recall und Precision gleich hoch ist. So wäre zum Beispiel das F-Maß für unser Rechenbeispiel aus Kapitel 3.8 (Precision = 0,734; Recall = 0,579) bei einem Gewicht β von 1:

$$F_1 = \frac{(1^2 + 1) * 0{,}579 * 0{,}734}{0{,}579 + 1^2 * 0{,}734} = \frac{2 * 0{,}579 * 0{,}734}{0{,}579 + 0{,}734} = 0{,}65$$

während sich analog dazu der Wert für das F-Maß bei einem Gewicht β von 2 nur noch auf 0,60 beläuft. Die Grafik in Abbildung 8 zeigt den Verlauf von F0,1 bis F2,0.

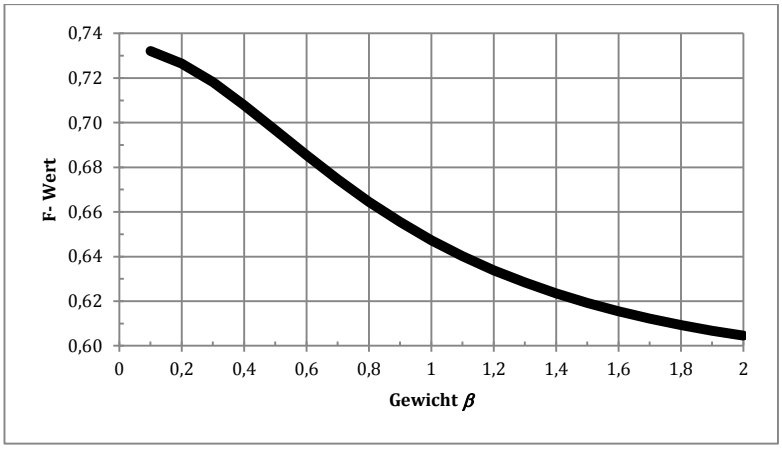

Abbildung 8: Verlauf des F-Maßes in Abhängigkeit des Gewichts β (Precision = 0,734; Recall = 0,579)

Wie in diesem Kapitel gezeigt, ermöglicht die Verwendung des F-Maßes bei der Evaluation von Informationssuchsystemen eine prägnante, aussagekräftige Beurteilung der Effektivität durch die Kombination der beiden fundamentalen Kennzahlen Precision und Recall. Des weiteren wurde gezeigt, wie mehrere Systeme auf dieser Grundlage miteinander verglichen werden können und wie durch den Einsatz von Gewichten die Aussagekraft des F-Maßes an die jeweiligen Evaluations- beziehungsweise Entwicklungsanforderungen angepasst werden kann. Da diese Kennzahl aber ausschließlich auf dem Konzept von Precision und Recall basiert, sieht sich auch das F-Maß selbst einer vergleichbaren Kritik ausgesetzt.

Powers nennt in seiner Arbeit (Powers 2011) zahlreiche Möglichkeiten für potentielle Verzerrungen und Abweichungen, denen man sich beim Einsatz von Precision, Recall und dem F-Maß bewusst sein muss. So bemängelt er vor allem, dass der Effektivität eines Systems im Umgang mit „True Negativ“ Elementen, also jenen Elementen, die korrekterweise vom System als nicht relevant beurteilt wurden, zu wenig Beachtung geschenkt wird. Weiters kritisiert er, dass die zugrunde liegenden Verteilungen und Vorurteile einen zu starken Einfluss auf das Ergebnis haben. In seiner Arbeit präsentiert er mehrere Verbesserungsvorschläge für spezifische Anwendungsfälle. Dennoch gehören das F-Maß so-

wie dessen zugrunde liegenden Kennzahlen Precision und Recall heute zu den am weitesten verbreiteten und anerkanntesten Evaluationsmetriken auf dem Gebiet der Informationssuche.

3.10 Bewertung von gereihten Suchergebnissen

Wie von Croft el al. in (Croft et al. 2010) beschrieben, liefern die meisten der heute existierenden Suchtechnologien die Elemente des Suchergebnisses als sortierte Liste zurück an den Aufrufer, wobei das Element mit der höchsten Relevanz an oberster Stelle der Liste steht. Die weiteren Elemente folgen in der Liste mit absteigend sortierter Relevanz. Dies entspricht einer Erweiterung des oben definierten Suchparadigmas („Ein Suchsystem soll so viele relevante Elemente wie möglich und dabei so wenig nicht relevante Elemente wie möglich liefern“, vgl. Kapitel 3.1) dahingehend, als dass ein Suchsystem möglichst viele relevante Elemente mit einem möglichst hohem Rang (d.h. möglichst weit vorne in der Ergebnisliste) zurück liefern soll. Diese Annahme basiert auf der Beobachtung, dass die Benutzer/innen eines Informationssuchsystems den zuoberst gereihten Elementen mehr Aufmerksamkeit schenken und diese daher die wichtigsten Elemente des Suchergebnisses darstellen.

Um diesem Umstand auch bei der Effektivitätsevaluation Rechnung zu tragen und die fundamentalen Bewertungsmetriken Precision und Recall (vgl. Kapitel 3.8) für gereihte Suchergebnisse anwenden zu können, werden die beiden Kennzahlen an jeder Position (d.h. an jedem Rang) innerhalb der gereihten Ergebnisliste separat berechnet. Die Kennzahlen werden also nicht für die gesamte Ergebnismenge ermittelt, sondern nur für die Teilmenge bis zur jeweiligen Position (vgl. Punkt (1) „Precision at Rank p“ unten). Die exemplarische Auflistung in Tabelle 8 zeigt für zwei unterschiedliche Suchergebnisse (*Ergebnis*$_1$, *Ergebnis*$_2$) die jeweils ersten zehn zurück gelieferten Elemente (*Position 1* ..

10) sowie die Relevanz jedes einzelnen Elements (*Relevant$_1$, Relevant$_2$*). Wie aus der untenstehenden Tabelle ersichtlich, beinhalten beide Ergebnislisten insgesamt sechs relevante und vier nicht relevante Dokumente. Somit sind auch die Werte für Precision und Recall an der Position zehn ident: *Relevant$_1$(10) = Relevant$_2$(10) = 1,00* (da alle sechs relevanten Dokumente gefunden wurden) und *Precision$_1$(10) = Precision$_2$(10) = 0,60* (da sechs der insgesamt zehn Elemente im Suchergebnis als relevant gelten). Die beiden Suchergebnisse können an dieser Stelle als gleich effektiv erachtet werden.
Betrachtet man aber die Kennzahlen an einem höheren Rang, wird schnell deutlich, dass sich die beiden Systeme sehr wohl unterscheiden. So beträgt zum Beispiel der *Recall* Wert von *Ergebnis$_1$* an der *Position 5 0,67* (das Suchergebnis beinhaltet an dieser Position vier von sechs relevanten Elementen), während sich der gleiche Wert von *Ergebnis$_2$* nur auf *0,33* (das Suchergebnis beinhaltet an dieser Position zwei von sechs relevanten Elementen) beläuft. Auch der *Precision* Wert von *Ergebnis$_1$* (*0,80*) übertrifft jenen von *Ergebnis$_2$* (*0,40*). *Ergebnis$_1$* erreicht also an *Position 5* eine höhere Effektivität als *Ergebnis$_2$*. Anhand dieses trivialen Beispiels lässt sich sehr einfach darstellen, welche Auswirkungen die Position (d.h. der Rang) eines Elements in der Ergebnisliste bei der Effektivitätsbewertung eines Suchsystems hat.

	Position	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Ergebnis$_1$	Relevant$_1$	J	N	J	J	J	J	N	N	N	J
	Recall$_1$	0,17	0,17	0,33	0,50	0,67	0,83	0,83	0,83	0,83	1,00
	Precision$_1$	1,00	0,50	0,67	0,75	0,80	0,83	0,71	0,63	0,56	0,60
Ergebnis$_2$	Relevant$_2$	N	J	N	N	J	J	J	N	J	J
	Recall$_2$	0,00	0,17	0,17	0,17	0,33	0,50	0,67	0,67	0,83	1,00
	Precision$_2$	0,00	0,50	0,33	0,25	0,40	0,50	0,57	0,50	0,56	0,60

Tabelle 8: Precision und Recall Werte der ersten zehn Ränge für zwei unterschiedliche Suchergebnisse bei insgesamt sechs relevanten Dokumenten; J ... Ja (relevant), N ... Nein (nicht relevant); in Anlehnung an (Croft et al. 2010), S315

Der Vergleich auf Grundlage separater Kennzahlen für jeden einzelnen Rang eignet sich jedoch nur bei Systemen mit sehr kleinen Ergebnislisten beziehungsweise Datensammlungen. Bei größeren Datenmengen wird diese Vorgehensweise jedoch sehr schnell unüberschaubar und impraktikabel. In der wissenschaftlichen Praxis werden daher Einzel-

ergebnisse bei der Effektivitätsbewertung häufig aggregiert. Dazu existieren in der Literatur verschiedene Ansätze. Croft et al. führen in ihrer Arbeit die folgenden drei grundsätzlichen Verfahren an:

(1) **Precision at Rank p.** Diese Methode wird in der Literatur oft auch mit „Precision at k“ oder „P@k“ bezeichnet und stellt die einfachste Form der Ergebnisverdichtung dar. Dabei werden schlichtweg ein oder mehrere bestimmte Ränge festgelegt, an denen die Werte für Precision berechnet beziehungsweise verglichen werden. Ein System A gilt dann als das bessere (d.h. effektivere) System, wenn es bei Rang p bessere Precision und Recall Werte aufweist als System B. Die Recall Werte brauchen nicht explizit verglichen zu werden, da generell gilt, dass ein an einer bestimmten Position höherer Precision Wert eines Systems A (im Vergleich zu System B) einen höheren Recall Wert bedingt. Prinzipiell kann der Wert für den Parameter p frei gewählt werden. In der wissenschaftlichen Praxis haben sich aber vor allem die Einstellungen 10 („Precision at 10“) und 20 („Precision at 20“) etabliert, da diese Werte die Anforderungen des alltäglichen Suchverhaltens („Bewerte die ersten 10/20 Treffer“) am ehesten widerspiegeln. „Precision at Rank k“ stellt eine unkomplizierte und sehr weitverbreitete Bewertungsmetrik dar. Es wird jedoch kritisiert, dass sich beim Einsatz dieser Metrik das übergeordnete Suchziel von „Finde möglichst viele relevante Dokumente“ hin zu „Finde möglichst viele relevante Dokumente innerhalb der ersten p Treffer“ verschiebt und die Systemeffektivität unterhalb des Ranges p ignoriert wird. Ferner wird bemängelt, dass innerhalb der ersten p Ränge keinerlei Unterscheidung getroffen wird. So würde beispielsweise ein Vergleich der beiden Suchergebnisse aus Tabelle 8 auf Basis von P@10 (Precision at Rank 10) ergeben, das beide Systeme gleich effektiv sind, während $Ergebnis_1$ auf Basis von P@5 deutlich besser abschneiden würde.

(2) **Standard Recall Levels.** Eine Alternative zu P@k ist die Berechnung von Precision Werten für einen vorgegebenen Bereich an Recall Werten *r*. Dieser Bereich erstreckt sich in der Regel von $0 \leq r \leq 1$ in einem Intervall von 0,1. Jedes Suchergebnis kann somit durch elf Precision-Recall Wertepaare repräsentiert werden.

Ein Vorteil gegenüber der P@k Methode ist dabei, dass das Ergebnis aller relevanten Dokumente (nicht nur der ersten k Elemente) berücksichtigt wird, da das gesamte Recall Spektrum (von 0 bis 1) in die Berechnung einfließt. Die Darstellung dieser Werte erfolgt oft in Form eines Recall-Precision Graphs (vgl. Kapitel 3.9, Abbildung 7). Ein häufiges Problem bei diesem Verfahren ist jedoch, dass – wie auch bei obigen Beispiel in Tabelle 8 – keine Precision Werte für die vorgegebenen Recall Schritte (0,1; 0,2; 0,3; ...) existieren. Diese Werte müssen dann mit Hilfe von Interpolationsverfahren rekonstruiert werden.

(3) **(Mean) Average Precision.** Die dritte und heute verbreitetste Methode zur Aggregation von separaten Precision-Werten gereihter Suchergebnisse ist die Berechnung von gemittelten Precision Werten. Dabei wird der arithmetische Mittelwert gebildet, indem für jeden Rang des Suchergebnisses, der ein relevantes Element enthält, der Precision Wert summiert und die Summe schlussendlich durch die Anzahl der relevanten Elemente im Suchergebnis dividiert wird. Nachstehende Formel zeigt die Berechnung der Average Precision für eine Menge relevanter Elemente R:

$$AP(R) = \frac{1}{|R|} * \sum_{j=1}^{|R|} Precision(r_j)$$

Zurückkommend auf das Beispiel in Tabelle 8 würde sich für Ergebnis1 eine Average Precision AP1 von

AP1 = (1,00 + 0,67 + 0,75 + 0,80 + 0,83 + 0,60) / 6 = 0,78

und für Ergebnis2 eine Average Precision AP2 von

AP2 = (0,50 + 0,40 + 0,50 + 0,57 + 0,56 + 0,60) / 6 = 0,52

ergeben. Wie aus obiger Berechnung erkenntlich, stellt Ergebnis1 auch unter Verwendung von Average Precision das effektivere System dar. Die Verwendung dieser Kennzahl birgt einige Vorteile im Vergleich zu den vorher genannten Methoden, da die gesamte Aussagekraft in einer einzigen Zahl gebündelt

wird. Zusätzlich werden alle relevanten Elemente im Suchergebnis berücksichtigt. Besonderes Gewicht liegt bei dieser Kennzahl auf den relevanten Suchergebnissen der obersten Ränge. Daher eignet es sich besonders zur Evaluation von Systemen, die darauf abzielen, dass die relevantesten Ergebnisse auch an den vordersten Listenplätzen aufscheinen. Laut der Dokumentation von Manning et al. (Manning et al. 2008) zählt diese Methode auch zu denjenigen mit der höchsten statistischen Diskriminierung und Stabilität.

Die obige Beschreibung der Average Precision sowie das angeführte Rechenbeispiel beziehen sich jedoch jeweils auf eine einzelne Suchanfrage. Für eine aussagekräftige Evaluation eines Suchsystems ist es aber nötig, eine Vielzahl von Suchanfragen zu bewerten und die Ergebnisse aller Suchanfragen in einen gemeinsamen Mittelwert zu aggregieren. Geschieht dies in Form des arithmetischen Mittelwerts (Summe aller Average Precision Werte durch die Anzahl dieser), so spricht man von Mean Average Precision (MAP). MAP ist die häufigste Form der Mittelwertbildung und berechnet sich für eine Menge an Suchergebnissen Q und eine zugehörige Menge an Average Precision AP Werten nach der Formel:

$$MAP(Q) = \frac{1}{|Q|} * \sum_{j=1}^{|Q|} AP_j$$

Eine alternative aber weniger gebräuchliche Variante ist der Geometric Mean Average Precision (GMAP), der vor allem Suchanfragen mit schlechterer Effektivität stärker zum Vorschein bringt.

In der wissenschaftlichen Gemeinde existieren noch zahlreiche weitere Metriken zur Bewertung der Effektivität von Informationssuchsystemen. Sie sind jedoch weniger weit verbreitetet als die oben angeführten Kennzahlen und werden häufig bei Systemen mit speziellen Evaluationsanforderungen angewandt, wie beispielsweise „Receiver Operating Characteristics“ (ROC) zur Analyse des vollständigen Suchspektrums, „(Normali-

zed) Discounted Cumulative Gain“ ((N)DCG) zur Evaluation auf Basis nicht-binärer Relevanzbeurteilungen oder „(Mean) Reciprocal Rank“ ([M]RR) zur Effektivitätsbewertung von Systemen mit nur einem relevanten Suchergebnis. Diese Metriken sind jedoch für den weiteren Verlauf der gegenständlichen Arbeit von keiner besonders großen Wichtigkeit. Für eine detaillierte Auseinandersetzung mit der Thematik sei der interessierte Leser aber auf die umfangreich vorhandene Literatur (Powers 2011; Raghavan et al. 1989; Manning et al. 2008; Sanderson 2010; Manning & Schuetze 1999; Croft et al. 2010) verwiesen.

4 Analyse von natürlich-sprachlichen Texten

Die maschinelle Analyse und Verarbeitung natürlicher Sprache mit Hilfe von computergestützten Systemen ist ein umfangreicher und vielschichtiger Bereich, dessen Verfahren und Erkenntnisse heute in einer Vielzahl von wissenschaftlichen und kommerziellen Forschungsdisziplinen und Anwendungsgebieten zum Einsatz kommen. Auch bei dem in dieser Arbeit dokumentierten Rollen-sensitiven Informationssuchsystem ROBUS werden Methoden und Technologien aus dem Bereich der Computerlinguistik intensiv genutzt. Sie werden im System vor allem zur automatisierten Generierung der textuellen Rollenprofile verwendet. Die Termvektor-basierten Profile sind für die Reihung und Gewichtung in ROBUS essentiell. Die Qualität der Suchergebnisse und damit des ganzen Systems hängt maßgeblich von diesen Rollenvektoren ab. Ohne den Einsatz von computerlinguistischen Methoden wäre die automatisierte Erstellung von Rollenprofilen in ROBUS nicht möglich.

Dieses Kapitel beginnt mit einer Einführung in das Gebiet der Computerlinguistik. Es werden alle wesentlichen Komponenten beleuchtet und eine Abgrenzung gegenüber anderen Forschungsdisziplinen vorgenommen. In weiterer Folge werden verschiedene grundlegende und aktuelle Methoden der Computerlinguistik beschrieben, wobei ein besonderer Schwerpunkt auf die für ROBUS relevanten Methoden gelegt wird. Abschließend werden mehrere Technologien und Systeme präsentiert, die den aktuellen Stand der Technik auf diesem Gebiet widerspiegeln und – direkt oder indirekt – für ROBUS verwendet werden.

4.1 Einführung

4.1.1 Wurzeln der automatisierten Sprachverarbeitung

Die menschlichen Bestrebungen, natürliche Sprache maschinell verarbeiten zu können, lassen sich bis in das 17. Jahrhundert zurückverfolgen. In dieser Zeit wurden bereits die ersten mechanischen Wörterbücher von Cave Beck (1657) und Johann Joachim Becher (1661) entwickelt. Auch die beiden 1933 vorgestellten Übersetzungsmaschinen von George Artsrouni (Mechanisches Wörterbuch für Zugfahrpläne) und Petr Petrovich Smirnov-Trojanskij (Dreiphasiger Translationsprozess mit einem mechanischem Wörterbuch zum Übersetzen der Grundformen und Funktionen in die Zielsprache) wurden noch vor Erfindung des ersten Computers[16] entwickelt. Die automatisierte maschinelle Übersetzung (engl. „Machine Translation“) von einer Quell- in eine Zielsprache war auch die Hauptmotivation von Warren Weaver. In seinem am 15. Juli 1949 veröffentlichten „Weaver Memorandum“[17] beschreibt er erstmals die Möglichkeit der „Übersetzung von einer Sprache in eine andere [...] durch den Einsatz von modernen Computersystemen mit sehr hoher Geschwindigkeit, Kapazität und logischer Flexibilität“ (Hutchins 1986).
Dieses Schreiben, das Weaver an (potentiell) interessierte Kollegen verschickte, gilt heute als Meilenstein in der Geschichte der maschinellen Übersetzung und hat zur raschen

[16] Als erste Computer werden heute die Systeme von Konrad Zuse (Z1, 1936), Howard H. Aiken (MARK 1, 1939), John von Neumann (EDVAC, 1944) und John Mauchly & John Presper Eckert (ENIAC, 1946) betrachtet: http://www.hnf.de/museum/die-erfindung-des-computers.html

[17] http://www.mt-archive.info/Weaver-1949.pdf

Etablierung als eigene Forschungsdisziplin mit Forschungsgruppen an mehreren renommierten Universitäten wie beispielsweise MIT[18], UCLA[19] und der UoW[20] geführt. Yehoshua Bar-Hillel wurde 1951 am MIT zum ersten Professor für Maschinelle Übersetzung ernannt und begann mit einer Studie zur aktuellen Lage des Themas. Im Jahr darauf wurde die erste englischsprachige und vier weitere Jahre darauf die erste internationale Konferenz am MIT veranstaltet. Im Jahre 1954 präsentiert Leon Dostert von der Georgetown University ein Pilotsystem zur bidirektionalen Übersetzung von russischen und englischen Texten, das in Kooperation mit der Firma IBM[21] entwickelt wurde.

Die Pionierjahre der maschinellen Übersetzung waren geprägt von großem wissenschaftlichen Enthusiasmus sowie von umfangreichen öffentlichen Fördergeldern. Die Ernüchterung kam 1966 in Form des ALPAC (Automatic Language Processing Advisory Committee) Reports, im Zuge dessen die Fortschritte der letzten Jahre als zu gering und die Kosten als zu hoch beurteilt wurden. Die Forschungsbestrebungen auf dem Gebiet der maschinellen Übersetzung galten vorerst als gescheitert, woraufhin auch die großzügigen Forschungsbudgets weitgehend gekürzt wurden. Stattdessen lautete die Empfehlung des ALPAC Reports, den Schwerpunkt in Zukunft auf die Erforschung der notwendigen linguistischen Grundlagen zu legen, da das Ziel einer automatisierten maschinellen Übersetzung ohne diese Grundlagen nicht zu erreichen ist. Die Computerlinguistik als eigenständige Forschungsdisziplin war geboren (Hutchins 1986).

---

[18] MIT ...Massachusetts Institute of Technology: http://www.mit.edu/

[19] UCLA ... University of California, Los Angeles: http://www.ucla.edu/

[20] UoW ... University of Washington: http://www.washington.edu/

[21] IBM ... International Business Machines Corporation: http://www.ibm.com/

4.1.2 Grundlagen der maschinellen Sprachverarbeitung

Die maschinelle Sprachverarbeitung oder Computerlinguistik ist ein umfangreiches Forschungs- und Entwicklungsgebiet, das in der zweiten Hälfte des 20. Jahrhunderts durch das Zusammenwirken von Linguistik (auch Sprachwissenschaft) und Informatik entstanden ist. Die Definitionen und Begrifflichkeiten in Bezug auf dieses Gebiet sind ebenso vielseitig wie deren Aufgabenschwerpunkte und Forschungsziele. Eine mögliche Kategorisierung beziehungsweise eine Übersicht der unterschiedlichen Teilbereiche geben Carstensen et al. (Carstensen et al. 2010). Dabei untergliedern sie den Gesamtbereich der Computerlinguistik anhand des Inhalts und der Ausrichtung (Theorie-orientiert vs. Praxis-orientiert) in die folgenden vier verschiedenen Themenfelder:

- Theoretische Computerlinguistik (engl. Computational Linguistics)
- Sprachtechnologie (engl. Human Language Technology)
- Linguistische Datenverarbeitung (Linguistic Computing)
- Natural Language Processing (NLP)

Unter dem Bereich der theoretischen Computerlinguistik verstehen Carstensen et al. jenen Bereich, der „für die maschinelle Sprachverarbeitung im Wesentlichen als Instrument zur Falsifizierung ihrer Theorien und Modelle eingesetzt wird“ während die Linguistische Datenverarbeitung „primär an der zweckorientierten Verarbeitung von Sprachdaten (Lexika, Korpora, etc.) interessiert ist“.
Des Weiteren nennen die Autoren das Gebiet des Natural Language Processings, welches „als Teilgebiet der Künstliche-Intelligenz-Forschung (KI) Sprache als ein kognitives Phänomen auffasst und zu simulieren versucht“. Der vierte Punkte definiert die Disziplin der Sprachtechnologie, „die in erster Linie an kommerziell einsetzbaren Sprachanwendungen interessiert ist“. Abbildung 9 zeigt die Unterteilung der Computerlinguistik in die vier beschriebenen Bereiche anhand ihrer wesentlichen Inhalte beziehungsweise Forschungsziele.

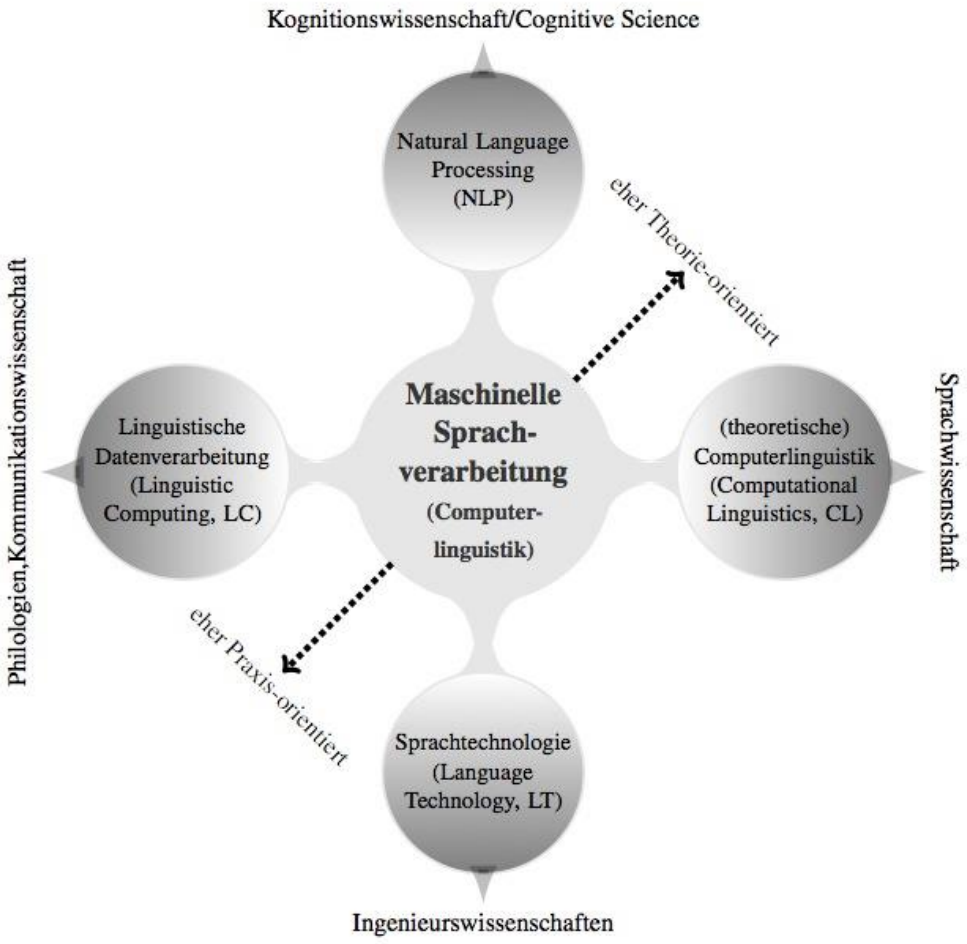

Abbildung 9: Unterteilung der Computerlinguistik in vier Bereiche nach Carstensen (Carstensen et al. 2010)

Der Begriff der Computerlinguistik wurde bereits Mitte der 60er Jahre durch David Hayes, einem Mitglied des ALPAC Komitees, als Mitautor des ALPAC Reports (vgl. Kapitel 4.1.1), geprägt (Mitkov 2004). Die meisten Mitglieder der wissenschaftlichen Gemeinde, verstehen die Computerlinguistik als interdisziplinäres Fach. So beschreiben auch Jurafsky et al. die Entstehung aus dem Zusammenwirken der Fachbereiche Sprachwissenschaften (theoretische Computerlinguistik), Computerwissenschaften (Natürliche Sprachverarbeitung), Elektrotechnik/Elektronik (Spracherkennung & -erzeugung) und Psychologie (computergestützte Psycholinguistik) (Jurafsky et al. 2000).

Wie aber in obiger Abbildung veranschaulicht, handelt es sich um ein sehr umfangreiches und vielschichtiges Forschungs- beziehungsweise Entwicklungsgebiet. Es ist daher wenig verwunderlich, dass zahlreiche, zum Teil sehr divergente Definitionen für den Begriff der Computerlinguistik existieren:

- „Die C. beschäftigt sich mit Theorien, Verfahren, Modellen, Systemen und Werkzeugen zur automatischen Verarbeitung von gesprochener und geschriebener Sprache. Sie ist über die Informationslinguistik mit der Informationswissenschaft verbunden, der sie Verfahren und Werkzeuge zur Überwindung von Sprachbarrieren für die internationale Kommunikation zuliefert.“ (Informationswissenschaft 2013)
- „Computerlinguistik untersucht, wie menschliche Sprache mit Computern verarbeitet und interpretiert werden kann. Sie erforscht die mathematischen und logischen Eigenschaften natürlicher Sprache und entwickelt algorithmische und statistische Verfahren zur automatischen Sprachverarbeitung.“ (Computerlinguistik 2013)
- „Computational linguistics (CL) is a discipline between linguistics and computer science which is concerned with the computational aspects of the human language faculty. It belongs to the cognitive sciences and overlaps with the field of artificial intelligence (AI), a branch of computer science aiming at computational models of human cognition. Computational linguistics has applied and theoretical components.“ (Uszkoreit 1996)
- „Computational linguistics is the scientific study of language from a computational perspective. Computational linguists are interested in providing computational models of various kinds of linguistic phenomena. These models may be "knowledge-based" ("hand-crafted") or "data-driven" ("statistical" or "empirical").“ (Sproat 2005)

4.1.3 Sprachliche Komponenten

Ungeachtet der vielen unterschiedlichen Zielsetzungen und Herangehensweisen im Bereich der Computerlinguistik gibt es mehrere grundlegende sprachliche Komponenten, die bei der Analyse beziehungsweise Verarbeitung von Sprache relevant sind. Dazu zählen laut Mitkov (Mitkov 2004):

- Phonologie (Lautlehre, engl.: Phonology)
- Morphologie (Wortgrammatik, engl.: Morphology)
- Lexikographie (Wortsammlung, engl.: Lexicography)
- Syntax (Satzgrammatik, engl.: Syntax)
- Semantik (Bedeutung, engl.: Semantics)
- Pragmatik (Verwendung, engl.: Pragmatics)

Da diese sprachlichen Komponenten die Grundlage für viele computerlinguistische Verfahren und Methoden darstellen, werden sie im Folgenden einzeln erläutert. Dabei wird jeweils auf die spezifische Relevanz in Bezug auf das gegenständliche ROBUS Verfahren eingegangen.

4.1.3.1 Phonologie

Unter Phonologie versteht man die „systematische Untersuchung von in der Sprache verwendeten Lauten und ihre Zusammensetzung in Silben, Wörtern und Phrasen“. Dementsprechend beschäftigt sich die Computerphonologie mit der Anwendung von computergestützten Methoden zur Verarbeitung von phonologischen Informationen (Bird & Ellison 1994).
Die menschliche Stimme ist prinzipiell in der Lage unendlich viele Laute zu erzeugen. In jeder gesprochenen Sprache lassen sich diese Laute zu einer deutlich kleineren Menge

von distinktiven Lautklassen – sogenannten Phonemen – abstrahieren. So gibt es beispielsweise im Deutschen viele Varianten, den Laut *r* auszusprechen (stark gerollt, leicht gerollt, nicht gerollt). Für die Bedeutung eines Wortes (z.B.: *R*auch) sind diese unterschiedlichen Aussprachen jedoch nicht relevant und können somit alle dem gleichen Phonem */r/* zugeordnet werden. Im Gegensatz dazu würde der Laut *l* statt *r* die Bedeutung des gleichen Wortes sehr wohl verändern (*L*auch), daher zählt dieser zu einem anderen Phonem (Bird 2004).

Während die Phonologie als Teil der Grammatik in der theoretischen Linguistik eine wichtige Rolle spielt, ist ihre Bedeutung in der Computerlinguistik eher untergeordnet. Hier bildet sie heute zusammen mit der Phonetik die Grundlage für diverse Methoden und Verfahren der Spracherkennung und -synthese (Taylor 2009). Da sich diese Arbeit auf die Analyse und Verarbeitung von natürlicher Sprache in Form von schriftlichem Text konzentriert, kommt der Phonetik keine spezielle Bedeutung zu. Der interessierte Leser sei hier auf die in großem Umfang existierende Literatur (z.B.: (Bird 2004; Bird & Ellison 1994; Mitkov 2004)) verwiesen.

4.1.3.2 Morphologie

Natürliche Sprache kann als System verstanden werden, das auf unterschiedlichen Ebenen betrachtet und hinsichtlich verschiedener Aspekte analysiert werden kann. Eine zentrale Komponente jeder Sprache ist das Wort. Der Begriff beziehungsweise das Konzept des Wortes ist jedoch - sowohl umgangssprachlich als auch wissenschaftlich gesehen – ein sehr vielschichtiges (Vater 2002). So umfasst beispielsweise alleine der Wortschatz der deutschen Sprache laut Duden (Anon 2006) circa 500.000 Wörter, von denen in etwa 75.000 zum so genannten Alltagswortschatz zählen. Des Weiteren unterliegt jede Sprache einer zeitlichen Änderung, indem laufend neue Wörter aufgenommen werden, während bestehende Wörter aus dem Sprachgebrauch verschwinden.

Die Morphologie als wissenschaftliche Disziplin beschäftigt sich mit der Analyse von Form und Struktur von Wörtern und beschreibt die grundlegenden Einheiten sowie Verfahren zur Bildung von Wörtern einer Sprache. Dieser Aspekt spielt eine zentrale Rolle für viele Methoden der Computerlinguistik im Allgemeinen und für das in dieser Arbeit

präsentierte ROBUS Verfahren im Besonderen (vgl. Kapitel 5.1.4). Daher wird das Konzept der Morphologie im Folgenden anhand der Arbeit von Trost (Trost 2004) ausführlich erläutert.

Wörter, als zentrale Komponenten der Sprache, werden in der Literatur als Morpheme bezeichnet. Wie in (Trost 2004) beschrieben, definiert die Sprachwissenschaft Morpheme als „die elementarste Einheit, der eine inhaltliche Bedeutung (Semantik) beziehungsweise eine grammatikalische Funktion zugeordnet werden kann". Er beschreibt Morpheme weiter als „abstrakte Entitäten, die grundlegende Eigenschaften, entweder in Form von semantischen Konzepten - wie beispielsweise *door (Tür), blue (blau)* oder *take (nehmen)* oder in Form von abstrakten Eigenschaften wie zum Beispiel die Vergangenheitsform oder Plural (Mehrzahl), repräsentieren". Die Wörter einer Sprache können auf Basis eines oder mehrerer Morpheme gebildet werden.

Die konkrete Umsetzung eines Morphems als Teil eines Wortes wird als Morph bezeichnet. In manchen Fällen ist das zugrundeliegende Morphem ident mit dem gebildeten Morph. So wird beispielsweise aus dem Morphem *door* das Morph *door* gebildet. Anders verhält es sich beim Morphem *take*, das sowohl dem Morph *take* als auch dem Morph *took* zu Grunde liegt. Hierbei handelt es sich um so genannte Allomorphe: zwei Morphe mit unterschiedlicher Form aber mit derselben Funktion (Bedeutung). Des Weiteren unterscheidet man zwischen freien Morphen, die selbstständig ein Wort bilden können, und zwischen gebundenen Morphen (Affixen), die nur in Kombination mit anderen Morphen ein Wort bilden können. So besteht zum Beispiel das Wort *doors* (Türen, Mz.) aus dem freien Morph *door* und dem Affix *s*. Im Vergleich zur Anzahl der Wörter einer Sprache, ist die Anzahl der Morphe verhältnismäßig klein. Heutige Sprachen verwenden im Durschnitt circa 10.000 unterschiedliche Morphe.

In der Sprachwissenschaft werden Wörter in verschiedene Klassen eingeteilt. Diese Klassen werden meistens als Wortarten (engl.: Part of Speech, PoS) oder seltener auch als syntaktische beziehungsweise grammatische Kategorien bezeichnet. Bis heute existiert kein allgemein gültiges Klassifikationsschema. Vielmehr hängt die konkrete Klassifizierung von unterschiedlichen Parametern (Sprache, Forschungsdisziplin, etc.) ab. Aus Perspektive der Computerlinguistik existieren drei besonders wichtige Wortarten:

- Nomen (Substantiv; engl.: Noun)

- Verb (Zeitwort; engl.: Verb)
- Adjektiv (Eigenschaftswort; engl.: Adjective)

Nomen repräsentieren konkrete und abstrakte Entitäten wie beispielsweise Gegenstände, Personen, Lebewesen und Gefühle. Sie können von Adjektiven näher beschrieben werden, indem spezielle Eigenschaften eines Nomens durch Angabe eines Adjektivs spezifiziert werden. Verben hingegen kennzeichnen Tätigkeiten, Vorgänge und Aktionen (Manning & Schuetze 1999).
Abhängig vom betrachteten Klassifizierungsschema existieren unterschiedliche weitere Wortarten wie beispielsweise Pronomen (Fürwort), Adverbien (Umstandswort), Artikel (Begleitwort), Präpositionen (Vorwort oder Verhältniswort), Numerale (Zahlwort) und Konjunktionen (Bindewort).
Die Vorgänge und Zusammenhänge zwischen den verschiedenen Komponenten der Wortformen werden in der Morphologie durch drei wesentliche Prozesse beschrieben:

- Flexion (auch: Beugung)
- Derivation
- Komposition

Bei der **Flexion** eines Wortes ändert sich weder die Wortart noch seine Bedeutung; die Änderung betrifft aber seine grammatikalische Funktion. Dies äußert sich oft durch das Hinzufügen von Affixen, kann jedoch auch durch andere Regeln bestimmt werden. Die Flexion kann nicht auf alle Wortarten angewandt werden. Dementsprechend unterscheidet man in der Literatur zwischen flektierbaren und nicht-flektierbaren Wortarten. In der deutschen Sprache zählen Adverbien, Interjektionen, Konjunktionen und Präpositionen zu den nicht-flektierbaren Wortarten. In der englischen Sprache sind darüber hinaus auch Artikel nicht-flektierbar.
Bei den flektierbaren Wortarten untergliedert man die Flexion wiederum anhand der betreffenden Wortart in zwei Unterklassen. So spricht man bei der Flexion von Verben von der Konjugation und bei Substantiven sowie Adjektiven von der Deklination. Die Kon-

jugation von Verben ist bei allen indoeuropäischen Sprachen von der Person, dem Numerus (Zahl), dem Modus, dem Tempus (Zeit) und dem Genus (Geschlecht) abhängig. Untenstehende Tabelle zeigt eine Auflistung der unterschiedlichen Formen für die Beispiele *gehen*, *laufen* und *lieben* (Bauer n.d.).

Eigenschaft	**Bezeichnung**	**Beispiel**
Person	1.Person	ich gehe
	2.Person	du gehst
	3.Person	er/sie/es geht
Numerus	Singular	ich gehe
	Dual (Zweizahl)	beide gehen (heute nicht mehr gebräuchlich)
	Plural	wir gehen
Modus	Indikativ	er läuft
	Konjunktiv	er laufe
Tempus	Präsens	ich gehe
	Präteritum	ich ging
	Perfekt	ich bin gegangen
	Plusquamperfekt	ich war gegangen
	Futur I	ich werde gehen
	Futur II	ich werde gegangen sein
Genus	Aktiv	ich liebe
	Passiv	ich werde geliebt

Tabelle 9: Beispiele zur Konjugation von Verben[22]

Bei der Deklination von Substantiven und Adjektiven sind die drei Parameter Genus (Geschlecht), Numerus (Zahl) und Kasus (Fall) ausschlaggebend für die Beugung von Worten. Tabelle 10 enthält Beispiele für unterschiedliche Beugungen in Abhängigkeit dieser drei Eigenschaften.

[22] http://www.uni-protokolle.de/Lexikon/Konjugation_(Grammatik).html

Genus	Kasus	Beispiel - Singular	Beispiel - Plural
Feminin (weiblich)	Nominativ	die grüne Wiese	die grünen Wiesen
	Genitiv	der grünen Wiese	der grünen Wiesen
	Dativ	der grünen Wiese	den grünen Wiesen
	Akkusativ	die grüne Wiese	die grünen Wiesen
Maskulin (männlich)	Nominativ	der große Baum	die großen Bäume
	Genitiv	des großen Baumes	der großen Bäume
	Dativ	dem großen Baum	den großen Bäumen
	Akkusativ	den großen Baum	die großen Bäume
Neutrum (sächlich)	Nominativ	das rote Auto	die roten Autos
	Genitiv	des roten Autos	der roten Autos
	Dativ	dem roten Auto	den roten Autos
	Akkusativ	das rote Auto	die roten Autos

Tabelle 10: Deklination von Adjektiven und Substantiven

Beim Vorgang der **Derivation** werden durch Hinzufügen von Präfixen oder Suffixen zur Grundform (Grundmorphem) gänzlich neue Wörter gebildet. Anders als bei der Flexion, können sich dabei aber nicht nur die grammatikalische Funktion, sondern die Wortart und die Bedeutung (Semantik) eines Wortes ändern. So wird beispielsweise aus dem deutschen Verb *lesen* durch Derivation mittels Anhängen des Suffixes *bar* das Adjektiv *les-bar*. Eine weitverbreitete Derivationsform im Englischen ist das Hinzufügen des Affixes *ly*. Wird dieses Nachmorphem zum Beispiel dem Adjektiv *hard* (hart, fest, schwer) angehängt, ändert sich sowohl die Wortart (von Adjektiv zu Adverb) als auch die Bedeutung: *hard-ly* = kaum, fast nicht, schwerlich.

Die Derivation ist weniger systematisch und vollständig als die Flexion, da sie nicht auf alle Grundformen mit den gleichen Regeln angewandt werden kann. So gibt es etwa im Deutschen zwar das Wort *hör-bar*, jedoch nicht das Wort *seh-bar*; stattdessen existiert aber der Begriff *sicht-bar* sehr wohl. Anhand des letzteren Beispiels lässt sich eine weitere Eigenschaft der Derivation – die Rekursion – zeigen: Derivate (Wörter, die durch Derivation gebildet wurden) können selbst als Grundform für einen neuerlichen Derivationsvorgang dienen. Aus dem Derivat *sicht-bar* kann somit durch Voranstellen des Präfixes *un-* das Adjektiv *un-sicht-bar* und daraus wiederum durch Hinzufügen des Suffixes *–keit* das Substantiv *Un-sicht-bar-keit* gebildet werden.

Beim Vorgang der **Komposition** werden - im Unterschied zur Derivation – die Grundformen von mindestens zwei unterschiedlichen Morphemen miteinander zu einem neuen Wort verbunden. So können beispielsweise die beiden deutschen Nomen *Haus* und *Boot* durch Komposition zu einem neuen Wort *Hausboot* vereint werden. An der Verbindungsstelle zwischen zwei Komponenten können in manchen Fällen, wie etwa in *Boot-s-haus*, zusätzliche Verbindungsmorphe vorkommen.
Die semantische Interpretation von Kompositionen ist äußerst schwierig, da (1) semantische Relationen zwischen den Komponenten nicht eindeutig zugeordnet werden und (2) eine strikte Abgrenzung von Kompositionen gegenüber Phrasen in vielen Fällen nicht möglich ist. Erstes Problem veranschaulicht Trost (Trost 2004) anhand des Schnitzel-Beispiels. Die Bedeutung eines Kompositums, dessen zweite Komponente das Nomen *Schnitzel* darstellt, ändert sich grundlegend in Abhängigkeit der eingesetzten ersten Komponente: *Wiener-schnitzel* („Schnitzel nach Wiener Art"), *Schweins-schnitzel* („Schnitzel aus Schweinefleisch"), *Kinder-schnitzel* („Schnitzel für Kinder"). Für das zweite Problem nennt selbiger das englische Beispiel *red wine*, das sowohl eine aus mehreren Wörtern bestehende Phrase als auch ein zusammengesetztes Kompositum darstellen kann.

4.1.3.3 Morphologische Merkmale von Stellenausschreibungstexten

Wie bereits eingangs erwähnt, dient die Morphologie als Grundlage für viele computerlinguistische Verfahren. Im Rahmen der gegenständlichen Arbeit ist sie insbesondere für die Reduktion von Wörtern auf ihre Stammform (engl.: Stemming), die Rückführung von Wörtern auf ihre normierte Grundform (Lemmatisierung) sowie für die Zuordnung von Wortarten (engl.: Part-of-Speech Tagging) von Relevanz.
Die im Kontext dieser Arbeit relevanten Stellenausschreibungstexte liegen dabei in englischer Sprache vor und weisen hinsichtlich ihres Inhalts einige Besonderheiten auf. Dazu zählen vor allem eine hohe Anzahl an branchenspezifischen Fachbegriffen und Eigennamen (Firmennamen, Produktbezeichnungen, etc.). Die nachstehende Auflistung zeigt drei Auszüge von unterschiedlichen Stellenausschreibungstexten des Online-Portals LinkedIn[23], die exemplarisch verdeutlichen sollen, dass (1) diese Art von Begriffen sehr häufig

[23] www.linkedin.com

vorkommt, und dass (2) diese Begriffe zudem eine große Bedeutung für den Inhalt des Textes darstellen:

- Textsegment einer Ausschreibung für die Funktion als Web Developer: "Apply front-end technologies including i.e. HTML, CSS, JavaScript. Maintain our Linux and Microsoft Windows Servers."
- Textsegment einer Ausschreibung für die Funktion als Financial Consultant: „Job responsibilities include strong knowledge of Integration of FI/CO, SD and MM including Procure to pay (P2P) and experience in using Accounting Principle/practices like IFRS/GAAP."
- Textsegment einer Ausschreibung für eine Funktion als SEO Specialist: „We are currently looking for a talented SEO Specialist to work at our Spencerport, NY location. [...] Desired Skills: Understanding of HTML/CSS development, experience with WordPress, Google Adwords and Google Analytics."

Ziel des ROBUS Verfahrens ist es, aus diesen Ausschreibungstexten möglichst viele rollenspezifische Terme zu extrahieren, die eine bestimmte Berufsgruppe (Rolle) distinktiv beschreiben und damit terminologisch zu einer eindeutigen Abgrenzung dieser Berufsgruppe beitragen. Zu diesen Begriffen zählen vorrangig Tätigkeitsbezeichnungen wie beispielsweise *Physicist* oder *Specialist*, besondere Fertigkeiten wie etwa Sprachkenntnisse (z.B.: *English, Spanish*) und Namen von Produkten, Firmen sowie Technologien (z.B.: *Microsoft, Logitech*).

Für die computerlinguistische Analyse der Ausschreibungstexte in ROBUS sind die morphologischen Eigenschaften jener Wörter von besonderem Interesse. So können bestimmte Affixe von Wörtern als morphologische Indikatoren interpretiert werden (Bick 2004). Mehrere konkrete Beispiele nennen Nadeau et al. (Nadeau & Sekine 2007) in ihrer Arbeit. Demnach beschreiben Wörter mit dem Suffix *-ist* im Englischen in vielen Fällen Berufsbezeichnungen (*Pshysic-ist, Special-ist*). Hingegen kennzeichnen Wörter die auf *–ish* oder *–an* enden häufig Sprachen und Nationalitäten (*Engli-sh, Germ-an*). Firmen- und Produktnamen wiederum verwenden oft die Suffixe *–tech* und *–soft* (*Micro-soft, Logi-tech*).

4.1.3.4 Lexikographie

Die Lexikographie befasst sich mit dem Aufbau, der Generierung und dem Einsatz von Lexika und definiert „Vorgang, Ergebnis und Methode der Anfertigung von Wörterbüchern“. Unter einem Lexikon wiederum versteht man die „Zusammenstellung der Wörter einer Sprache (bzw. eines regionalen, soziolektalen oder fachspezifischen Ausschnitts) in alphabetischer oder begrifflicher Ordnung zum Zwecke des Nachschlagens“ (Bussmann 2002).

Die Computerlexikographie im Speziellen untersucht einerseits Einsatzgebiete von Wörterbüchern in Computerprogrammen und andererseits die Verwendung von computergestützten Methoden zur Erstellung von neuen Wörterbüchern. Im Bereich der natürlichen Sprachverarbeitung werden Lexika heute von verschiedensten Applikationen verwendet. Dabei gilt zu beachten, dass derartige Wortsammlungen in der Regel für ihre jeweiligen Anwendungsfälle angepasst werden und keine vollständige beziehungsweise allumfassende Repräsentation darstellen. Vielmehr werden sie für ihr jeweiliges Einsatzgebiet wie etwa die Spracherkennung, die Informationssuche oder die maschinelle Übersetzung optimiert. Die Erstellung eines vollständigen Wörterbuchs einer lebenden Sprache wird in der Literatur schon deshalb als nicht möglich betrachtet, da bei einer lebenden Sprache kontinuierlich neue Wörter entstehen, während andere Wörter aus dem Sprachgebrauch verschwinden. Dies gilt in gleicher Weise auch für Wörterbücher, die einen bestimmten thematischen Ausschnitt (z.B. einen Fachbereich) abdecken (Hanks 2004).

Neben der sich daraus zwangsläufig ergebenden Unvollständigkeit von Wörterbüchern nennt Hanks (Hanks 2004) als weitere Hürde für den Einsatz solcher Wortsammlungen in computerlinguistischen Verfahren die unterschiedliche Handhabung von Eigennamen. Während etwa Orts- und Personennamen sowie Firmen- oder Produktbezeichnungen in manchen Lexika generell nicht berücksichtigt werden, finden sie in anderen Wörterbüchern Eingang, sofern sie von besonderer kultureller oder sprachlicher Relevanz sind. So enthält beispielsweise die Online-Ausgabe des Dudens (Duden 2013) sehr wohl einen Eintrag für den Firmennamen des Klebebandherstellers *Tixo*[24], während alle anderen Herstellerfirmen (z.B. *Tesa, Scotch, 3M*) nicht im Nachschlagewerk repräsentiert sind.

[24] http://www.duden.de/suchen/dudenonline/tixo

Trotz aller oben genannten Mängel und Einschränkungen spielen lexikalische Ressourcen eine bedeutende Rolle für viele computerlinguistische Methoden und die darauf aufbauenden Applikationen. In Bezug auf das in dieser Arbeit dokumentierte ROBUS Verfahren ist vor allem die Zuordnung von Wortbedeutungen mit Hilfe von Thesauri als relevanter Einsatzbereich hervorzuheben: ROBUS nutzt einen domänenspezifischen, multilingualen Thesaurus, um die Bedeutung von in Stellenausschreibungstexten gefundenen Nomina in Hinblick auf Fähigkeits- und Kompetenzbezeichnungen zu klassifizieren (vgl. Kapitel 5.1.5).

4.2 Einsatz von Thesauri für computerlinguistische Verfahren

Thesauri spielen eine wichtige Rolle in der Computerlinguistik und finden in vielen Verfahren und Applikationen, wie beispielsweise bei der Disambiguierung von Wortbedeutungen, Verwendung (Navigli 2009).

Sowohl die Verwendung des Begriffs *Thesaurus* an sich als auch das Verständnis über die Merkmale und Inhalte eines Thesaurus variieren aber in der Literatur sehr stark. Einig sind sich die meisten Autoren zumindest darüber, dass es sich dabei um ein wohldefiniertes, strukturiertes Vokabular handelt, welches verschiedenartige Relationen bereitstellt, um die enthaltenen Einträge miteinander in Verbindung zu setzen (Roth 2006).

(Kilgarriff & Yallop 2000) geben in ihrer Arbeit eine sehr allgemeine Definition ("[A thesaurus is] a resource in which words with similar meanings are grouped together") und zeigen auf, dass verschiedene Arten von Thesauri für verschiedene Anwendungszwecke existieren. Ferner beschreiben sie die wichtigsten Unterschiede zwischen Thesauri und Wörterbüchern. Der offensichtlichste Unterschied zwischen diesen beiden Ressourcentypen liegt in der Indexierung: Die Einträge von Wörterbüchern liegen in alphabetisch sortierter Form vor, während die Wörter in Thesauri nach ihrer Bedeutung gegliedert sind.

Kilgarriff und Yallop betonen, dass dies zwar das bekannteste, aber bei weitem nicht das einzige oder gar wichtigste Differenzierungsmerkmal ist: „If this were the only difference, a computational environment that offered both indexing possibilities would remove the distinction, and a resource [...], which offers both options, would be equally dictionary and thesaurus."

Ein weiterer wichtiger Unterschied besteht laut den Autoren darin, dass die meisten Thesauri – im Gegensatz zu Wörterbüchern – keine Definitionen der beinhalteten Wörter bereitstellen und dass die Gruppierung der Wörter nicht anhand expliziter semantischer Kategorien erfolgt. So ist für die Eintragung von Wörtern in Wörterbüchern die Existenz von distinktiven Bedeutungen und Gebrauchsweisen ausschlaggebend. Homonyme und Polyseme (Wörter die bei gleicher Schreibweise unterschiedliche Bedeutungen aufweisen) sind in Wörterbüchern explizit ausgewiesen. Eine Information, die in den meisten Thesauri mangels fehlender Definition des Eintrags verloren geht. Im Umkehrschluss kommt es häufig vor, dass ein Wörterbucheintrag an mehreren unterschiedlichen Stellen im Thesaurus gefunden werden kann (Kilgarriff & Yallop 2000).

Kilgarriff und Yallop unterscheiden in ihrer Arbeit zwischen vier verschiedenen Arten von Thesauri:

- Rogets Thesaurus (Thesauri nach Roget)
- WordNet / EuroWordNet
- Manuell erstellte Thesauri zum Einsatz in Informationssuchsystemen (Information Retrieval Systems)
- Automatisiert generierte, Korpus-basierte Thesauri

4.2.1 Rogets Thesaurus

Der Englische Mediziner Peter Mark Roget[25] hat 1852 den heute unter seinem Namen bekannten *Thesaurus of English Words and Phrases* veröffentlicht. Er wurde bereits 1957 im Bereich der automatisierten Sprachverarbeitung eingesetzt (Masterman 1957) und zählt nach wie vor zu einer der bedeutendsten Sammlungen auf diesem Gebiet.
Roget selbst beschreibt sein Werk als „... a collection of the words [the English language] contains and of the idiomatic combinations peculiar to it, arranged, not in alphabetical order as they are in a Dictionary, but according to the ideas which they express"; (Roget 1852) zitiert in (Jarmasz 2012). Der Thesaurus enthielt ursprünglich 15.000 Wörter und Phrasen aus fünf unterschiedlichen Wortklassen (Nomen, Verben, Adjektive, Adverbien und Interjektionen). Alle Einträge waren in sechs Klassen gegliedert:

- Abstract Relations (Abstrakte Beziehungen)
- Space (Raum)
- Material World (Materielle Welt)
- Intellect (Intellekt)
- Volition (Wille)
- Sentient and Moral Powers (empfindende und moralische Kräfte)

Die Einträge innerhalb jeder Klasse wurden weiter in Sektionen und Kategorien, und diese wiederum in Kopfeinträge, unterteilt. Der Umfang der enthaltenen Wörter und Phrasen wurde seit der Erstveröffentlichung mehrmals erweitert, während sich das Klassifikationsschema erstaunlicherweise nur marginal verändert hat (Jarmasz & Szpakowicz 2001).

[25] http://www.altiusdirectory.com/Society/thesaurus-day.php

9. Relation.

N. relation, bearing, reference, connection, concern,. cognation ; correlation &c. **12**; analogy; similarity &c. **17**; affinity, homology, alliance, homogeneity, association; approximation &c. (nearness) **197**; filiation &c. (consanguinity) **11**[obs3]; interest; relevancy &c. **23**; dependency, relationship, relative position.
comparison &c. **464**; ratio, proportion. link, tie, bond of union.
V. be related &c. adj.; have a relation &c. n.; relate to, refer to; bear upon, regard, concern, touch, affect, have to do with; pertain to, belong to, appertain to; answer to; interest.
bring into relation with, bring to bear upon; connect, associate, draw a parallel; link &c. **43**.
Adj. relative; correlative &c. **12**; cognate; relating to &c. v.; relative to, in relation with, referable or referrible to[obs3]; belonging to &c. v.; appurtenant to, in common with.
related, connected; implicated, associated, affiliated, allied to; en rapport, in touch with.
approximative[obs3], approximating; proportional, proportionate, proportionable; allusive, comparable.
in the same category &c. **75**; like &c. **17**; relevant &c. (apt) **23**; applicable, equiparant[obs3].
Adv. relatively &c. adj.; pertinently &c. **23**.
thereof; as to, as for, as respects, as regards; about; concerning &c. v.; anent; relating to, as relates to; with relation, with reference to, with respect to,with regard to; in respect of; while speaking of, a propos of[Fr]; in connection with; by the way, by the by; whereas; for as much as, in as much as; in point of, as far as; on the part of, on the score of; quoad hoc[Lat]; pro re nata[Lat]; under the head of &c. (class) **75** of; in the matter of, in re.
Phr. " thereby hangs a tale " [Taming of the Shrew].

Abbildung 10: Exemplarische Darstellung des Eintrags *Relation* aus Rogets Thesaurus (Anon 1991)

Obenstehende Abbildung zeigt ein Beispiel für den Eintrag *Relation* in Rogets Thesaurus (Version von 1911; online verfügbar unter (Anon 1991)). Wie aus der Abbildung ersichtlich, weist der Eintrag keine Definition des Begriffs beziehungsweise seiner Bedeutung(en) auf. Dafür enthält er jedoch eine ganze Reihe von Wörtern, die über verschiedene implizite Relationen mit dem Eintrag in Verbindung stehen und nach der Wortart (*N.* Nomen, *V.* Verb, *Adj.* Adjektiv, *Adv.* Adverb) gruppiert sind.

4.2.2 WordNet

Die lexikalische Datenbank WordNet wurde ab Mitte der 1980er Jahre von George A. Miller an der Princeton University entwickelt und ist heute sowohl für akademische als auch kommerzielle Zwecke frei verfügbar. Der Inhalt kann entweder direkt im Internet online betrachtet[26] oder heruntergeladen[27] werden.

WordNet besteht aus Konzepten, sogenannten Synsets, die in Form einer Menge von Synonymen repräsentiert werden. Jede Synonymgruppe repräsentiert genau ein Konzept. Ähnlich wie die Thesauri nach Roget (siehe oben) unterscheidet auch WordNet zwischen den vier Wortarten *Noun* (Nomen), *Verb* (Verb), *Adjective* (Adjektiv) und *Adverb* (Adverb), die im WordNet Umfeld auch oft als syntaktischen Kategorien bezeichnet werden. Interjektionen werden jedoch nicht (wie in Roget) separat berücksichtigt (Miller et al. 1990).

[26] WordNet Browser: http://wordnetweb.princeton.edu/perl/webwn

[27] WordNet Download: http://wordnet.princeton.edu/wordnet/download/current-version/

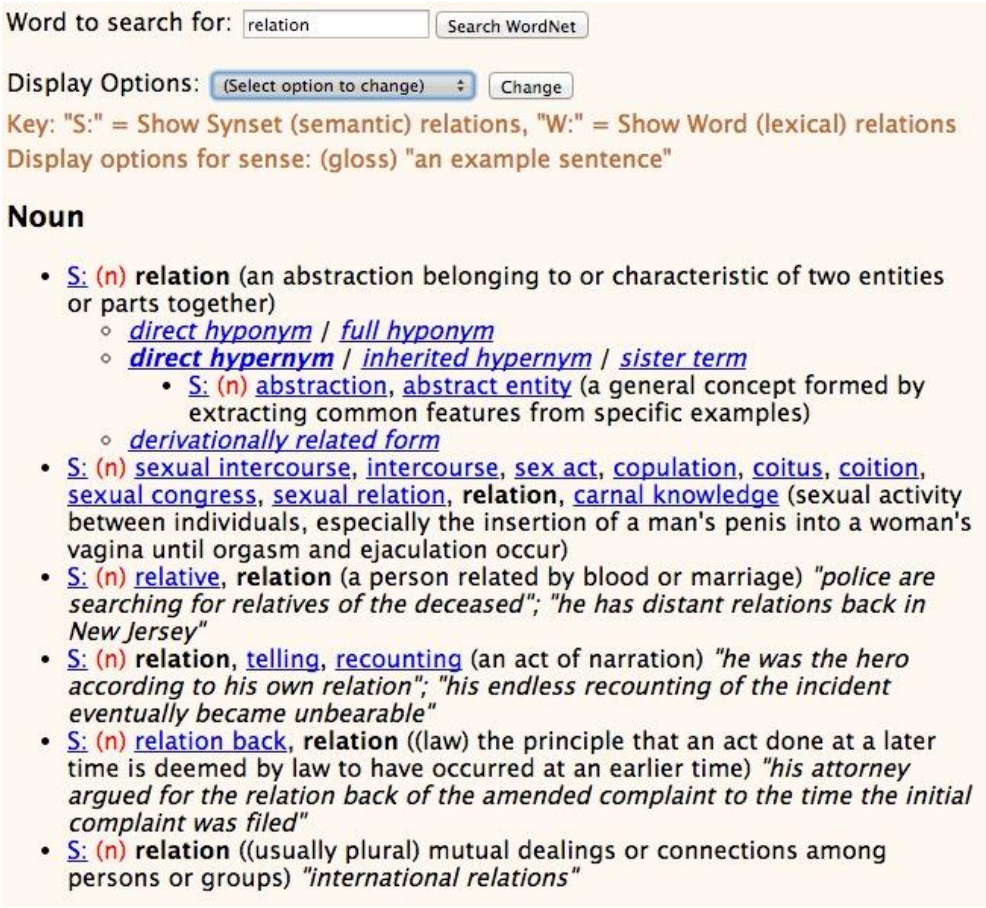

Abbildung 11: Exemplarische Darstellung des Eintrags *Relation* mit expliziter Beziehung (Hypernym) aus WordNet (Fellbaum 2012)

Das wesentliche Differenzierungsmerkmal zu den Thesauri nach Roget besteht allerdings in der Existenz von mehreren semantischen Relationen, die für die jeweiligen Wortarten expliziert definiert werden. Dabei unterscheidet die Datenbank zwischen sechs wesentlichen Relationstypen (Miller 1995):

- *Synonymie* (engl.: Synonymy): Die Bedeutung eines Wortes wird durch die Gruppierung mehrerer Synonyme (z.B.: Stiege - Treppe) in einem sogenannten Synset ausgedrückt. Es ist der bedeutendste und am häufigsten verwendete Relationstyp in WordNet. Es handelt sich dabei um eine symmetrische semantische Beziehung, die bei allen vier berücksichtigten Wortarten (Nomina, Verben, Adjektive und Adverbien) vorkommt.

- *Antonymie* (engl.: Antonymy): Dieser symmetrische Relationstyp wird hauptsächlich bei Adjektiven und Adverbien verwendet, um eine gegensätzliche Bedeutung (z.B.: kalt – warm) auszudrücken.
- *Hyponymie* (engl.: Hyponymy): Diese Relation wird verwendet, um auszudrücken, dass ein Synset (das Hyponym) eine untergeordnete beziehungsweise speziellere Bedeutung eines übergeordneten beziehungsweise allgemeineren Synsets (das Hypernym) darstellt. Diese Beziehungsart wird in der Literatur oft als Is-A Relation bezeichnet und kommt häufig bei Taxonomien zur Abbildung von Strukturbäumen für Nomina zum Einsatz. Auch in WordNet wird diese Relationsart ausschließlich zur Strukturierung von Nomina verwendet.
- *Meronymie* (engl.: Meronymy): Die Meronymie ist ebenfalls eine hierarchische Relation, die zur Abbildung von *Teil-Ganzes* (engl.: Part-Of) Beziehungen eingesetzt wird. Sie drückt aus, dass ein bestimmtes Konzept (das Meronym) ein Teil eines anderen Konzepts (das Holonym) ist. Das Holonym besteht in der Regel aus mehreren unterschiedlichen Meronymen; Beispiel: Blatt ist Meronym von Baum. Diese Beziehungsart ist in WordNet ebenfalls nur für Nomina definiert.
- *Troponomie* (engl.: Troponymy): Manche Tätigkeiten und Handlungen, die durch ein Verb ausgedrückt werden, können hinsichtlich der Art und Weise wie sie ausgeführt werden, spezifischer beschrieben werden. So ist beispielsweise *jodeln* eine spezielle Form von *singen*. Die beiden Konzepte können daher mittels Troponymie verbunden werden. Diese Relationsart ist in WordNet nur für Verben vorhanden und erfüllt für diese die gleiche Rolle, wie die Hyponymie für Nomina.
- *Implikation* (engl.: Entailment): Die Implikationsrelation bei Verben entspricht der Meronymie bei Nomina und gibt an, dass die Verwendung eines Verbs durch die Existenz eines anderen Verbs bedingt wird. Dies ist dann der Fall, wenn beispielsweise eine Tätigkeit oder Handlung Teil einer anderen Tätigkeit oder Handlung ist.
-

Semantic Relation	Syntactic Category	Examples
Synonymy (similar)	N, V, Aj, Av	pipe, tube rise, ascend sad, unhappy rapidly, speedily
Antonymy (opposite)	Aj, Av, (N, V)	wet, dry powerful, powerless friendly, unfriendly rapidly, slowly
Hyponymy (subordinate)	N	sugar maple, maple maple, tree tree, plant
Meronymy (part)	N	brim, hat gin, martini ship, fleet
Troponomy (manner)	V	march, walk whisper, speak
Entailment	V	drive, ride divorce, marry
Note: *N = Nouns* *Aj = Adjectives* *V = Verbs* *Av = Adverbs*		

Abbildung 12: Darstellung der explizit definierten semantischen Relationen (engl.: Semantic Relation) in WordNet mit zugehörigen Wortklassen (engl.: Syntactic Category) und Beispielen (engl.: Examples) (Miller 1995)

WordNet selbst ist nur für die englische Sprache verfügbar. Es wurden aber mittlerweile mehrere ähnliche Ressourcen für andere Sprachen auf den Prinzipien von WordNet entwickelt. Dazu zählen etwa GermaNet für das Deutsche (Hamp & Feldweg 1997) oder EuroWordNet (Vossen 2004), ein multilinguales Wortnetz, das Ende der 90er Jahre als Projekt der Europäischen Union geschaffen wurde und neben dem Deutschen GermaNet noch weitere Sprachen wie Spanisch, Niederländisch, Italienisch, Englisch, Tschechisch, Estnisch und Französisch integriert.

4.2.3 Manuell erstellte Thesauri

Im Laufe der Jahre wurden viele Informationssuchsysteme gezielt für die Anwendung in einem bestimmten Fachbereich entwickelt beziehungsweise adaptiert. Für diese Systeme haben domänenspezifische Thesauri eine besonders hohe Relevanz. Ihre Verwendung ermöglicht es den Informationssystemen, die Suchanfragen mithilfe der abgebildeten Taxonomien einzugrenzen oder auszuweiten. Des Weiteren können die Synonymrelationen in den Thesauri ausgenutzt werden, um bei der Suche nicht nur die eingegebenen Stichwörter sondern auch deren definierte Synonyme zu berücksichtigen (Kilgarriff & Yallop 2000).
Die Entwicklung dieser speziellen Art von Thesauri für den gezielten Einsatz in Informationssuchsystemen geht bis in die 1990er Jahre zurück. So beschreiben bereits Baeza-Yates und Ribeiro-Neto verschiedene Einsatzszenarien und definieren als die drei wesentlichen Relationsarten zwischen den Konzepten (Baeza-Yates & Ribeiro-Neto 1999):

- Engerer Ausdruck (engl.: Narrower Term)
- Weiterer Ausdruck (engl.: Broader Term)
- Verwandter Ausdruck (engl.: Related Term)

Als konkretes Beispiel für einen solchen Thesaurus nennen Kilgarriff und Yallop *The Unified Medical Language System (UMLS)*. Diese Sammlung aus dem biomedizinischen Bereich enthält 900.000 domänenspezifische Konzepte mit über 2 Millionen Synonymen, die durch 12 Millionen Relationen miteinander verbunden sind. Eine Besonderheit dieser, von der US amerikanischen *National Library of Medicine* entwickelten, Wortsammlung

besteht darin, dass sie 60 verschiedene Vokabulare, darunter die *National Center for Biotechnology Taxonomy*[28], die *Medical Subject Headings*[29], *Online Mendelian Inheritance in Man*[30] und die *Digital Anatomist Symbolic Knowledge Base*[31] in einer zentralen Datenbank integriert. Zudem können Relationen nicht nur zwischen den Konzepten innerhalb des Thesaurus, sondern auch zu Inhalten externer Ressourcen wie etwa der *GenBank*[32] spezifiziert werden (Bodenreider 2004).

Der wesentliche Unterschied dieser Thesauri zu den beiden oben genannten (Rogets Thesaurus, WordNet / EuroWordNet) liegt laut (Kilgarriff & Yallop 2000) darin, dass sie nicht danach trachten, eine möglichst umfassende, allgemeinsprachliche Ressource darzustellen, sondern – ganz im Gegenteil – für eine bestimmte Domäne erstellt werden und dementsprechend über ein fachspezifisches Vokabular und Set von Relationstypen verfügen. Domänenspezifische Informationssuchsysteme profitieren von diesem Umstand, da sie bei der Bearbeitung von Suchanfragen gezielt die Wörter und Beziehungen (Synonymie, Hyponymie) des abgebildeten Fachbereichs nutzen können. Es muss jedoch beachtet werden, dass solche Systeme von der Qualität des Thesaurus abhängig sind, und dass die Erstellung solcher Thesauri mit hohem Kosteneinsatz und Aufwand verbunden sein kann.

---

[28] http://www.ncbi.nlm.nih.gov/taxonomy

[29] http://www.nlm.nih.gov/mesh/

[30] http://www.omim.org/

[31] http://sig.biostr.washington.edu/~onard/AMIApapers/KBpaper.pdf

[32] http://www.ncbi.nlm.nih.gov/genbank/

4.2.4 Automatisiert generierte, Korpus-basierte Thesauri

Durch den Einsatz von verschiedenen computergestützten Methoden und Verfahren können Thesauri automatisiert erstellt und somit der manuelle Aufwand gänzlich vermieden beziehungsweise auf ein Minimum reduziert werden. Grundlage zur automatisierten Erstellung solcher Thesauri sind maschinenlesbare Textsammlungen (Korpora), auf Basis derer dann verschiedene Strategien zur Extraktion der Thesaurusinhalte angewandt werden können. Die einfachste und grundlegendste Strategie zur Umsetzung dieses Vorgangs besteht laut (Kilgarriff & Yallop 2000) darin, für jedes einzelne, im Korpus existierende Wort zu zählen, wie oft dieses *gemeinsam* mit jedem anderen Wort, das im Korpus enthalten ist, vorkommt. Dabei ist definiert, dass zwei Wörter dann *gemeinsam* vorkommen, wenn sie innerhalb einer Distanz *k* (beispielsweise innerhalb von vier (*k = 4*) Wörtern) im Korpustext aufscheinen. Nachfolgende Abbildung zeigt die Umsetzung dieser Strategie in Form von Pseudo-Code.

```
For each content word in the corpus
  for each other content word,
    find how often both occur within k
    words (or characters) of each other.
```

Abbildung 13: Vereinfachte Pseudocode-Darstellung einer automatisierten Korpus-basierten Thesaurusextraktion (Kilgarriff & Yallop 2000)

Durch die Anwendung dieser Strategie kann jedes Wort im Korpus als *n*-stelliger Vektor repräsentiert werden, wobei *n* der Gesamtanzahl aller Wörter im Korpus und der jeweilige Wert der Anzahl der gemeinsamem Vorkommen entspricht. Die unterschiedlichen Wörter können dann mithilfe dieses Vektors hinsichtlich ihrer Ähnlichkeit bewertet und dementsprechend in den Korpus integriert werden (Kilgarriff & Yallop 2000).
Als konkretes Beispiel für die Umsetzung eines solchen Thesaurus sei hier die Arbeit von (Castilho et al. 2012) angeführt: Die Autoren verwenden darin ein domänen-spezifisches Korpus aus dem Bereich Datenschutz und Privatsphäre. Dieses Korpus enthält 100 Gesetzestexte in englischer Sprache aus verschiedensten Ländern, darunter zum Beispiel das österreichische *Bundesgesetz über den Schutz personenbezogener Daten* (engl.: *Federal*

Act Concerning the Protection of Personal Data) oder das US-amerikanische *Gesetz zur Informationsfreiheit* (engl.: *Freedom of Information Act)*. Die Korpustexte stehen in gesammelter Form online zur Verfügung[33].

Castilho et al. verwenden in ihrer Arbeit drei verschiedene Thesaurusgenerierungsmethoden, um aus dem oben angeführten Korpus zu einer Menge von gegebenen Ausgangswörtern ähnliche Wörter automatisch zu extrahieren. Die auf diese Weise gefundenen Wörter werden in weiterer Folge zur Anreicherung einer domänen-spezifischen Ontologie genutzt. Die drei eingesetzten Vorgehensweisen werden in kombinierter Form verwendet und folgen dabei den folgenden unterschiedlichen Methoden (Castilho et al. 2012):

- *Grefenstette*: Analyse von syntaktischen Kontexten (Wörter oder Wortgruppen, die mit einem anderen Wort in einer syntaktischen Beziehung stehen) zur Bestimmung von Ähnlichkeiten zwischen Wörtern
- *Kaji*: Extraktion von Thesaurusinhalten anhand verschiedener statistischer Methoden (Termextraktion mittels Stoppwort-Filterung sowie Kookkurrenz- und Korrelationsanalyse)
- *Yang und Powers:* verwendet zusätzlich zur Methode nach *Grefenstette* das Verfahren der *Latent Semantic Analysis (LSA)*, um semantische Relationen zwischen Wörtern zu identifizieren

[33] Privacy Project – Corpus Visualization: http://www.cpca.pucrs.br/VisualizationTool/Resource/Corpus.html

4.3 Der DISCO Thesaurus

Der DISCO[34] Thesaurus ist eine multilinguale, domänenspezifische Wortsammlung und umfasst in der aktuellen Version über 100.000 Worteinträge in elf europäischen Sprachen. Der Inhalt des Thesaurus enthält ausschließlich Begriffe, die spezielle Fähigkeiten, Qualifikationen und Kenntnisse der Berufswelt definieren. Die interne Struktur der Einträge wird mittels Hyponomie und Hyperonomie als Taxonomiebaum abgebildet. Konkret definiert der Thesaurus drei explizite semantische Relationstypen (Müller-Riedlhuber 2009):

- *Semantische Äquivalenz*: jeder Eintrag im DISCO Thesaurus ist anhand seines einen eindeutigen primären Namen (Deskriptor) spezifiziert und kann zusätzlich eine oder mehrere alternative Bezeichnungen (Synonyme) aufweisen
- *Hierarchische Beziehung*: die hierarchische Ausprägung der Begriffe basiert auf dem Prinzip von „Gattung-zu-Art“ bzw. „Ganzes-zu-Teil“
- *Assoziative Beziehung*: verwandte Begriffe stehen miteinander in einer kontextuellen, semantischen oder benutzungsorientierten Beziehung

Der gesamte Inhalt des DISCO Thesaurus ist in zwei voneinander getrennte Taxonomiebäume aufgeteilt:

- *Fachliche Fertigkeiten und Kompetenzen*: Dieser Teil enthält Fähigkeits- und Qualifikationsbezeichnungen, die einem speziellen Fachgebiet (einer Domäne) zugeordnet werden können; er stellt somit den domänenspezifischen Teil dar. DISCO unterscheidet in diesem Teil 25 verschiedene Fachgebietskategorien (zum Beispiel *Architektur und Bauwesen, Bildung, Elektrotechnik*, etc.), die ihrerseits wiederum mehrere Unterkategorien haben. Abbildung 14 zeigt einen Ausschnitt

[34] European Dictionary of Skills and Competences

des domänenspezifischen Teils des DISCO Thesaurus Explorers, der online[35] zugänglich ist.

- *Überfachliche Fertigkeiten und Kompetenzen*: Dieser Teil der Taxonomie enthält alle fachübergreifenden, nicht domänenspezifischen Einträge wie beispielsweise Sprachkenntnisse, Führerscheinklassen und persönliche Eigenschaften. Der überfachliche Bereich ist in neun Fachgebietskategorien unterteilt. Auch der Inhalt des überfachlichen Bereichs ist mit Hilfe des Thesaurus Explorers auf der DISCO Projektseite[36] frei zugänglich. Die neun Hauptkategorien sowie alle Einträge der Kategorie *Führungs- und Organisationsfähigkeit* werden in Abbildung 15 dargestellt. Wie aus dieser Abbildung ersichtlich ist, ist der überfachliche (nicht domänenspezifische) Taxonomieteil sowohl hinsichtlich Umfang als auch Tiefe geringer, als der fachliche (domänenspezifische) Teil von DISCO.

[35] http://www.disco-tools.eu/disco2_portal/terms.php

[36] http://disco-tools.eu/disco2_portal/termSearchResult.php

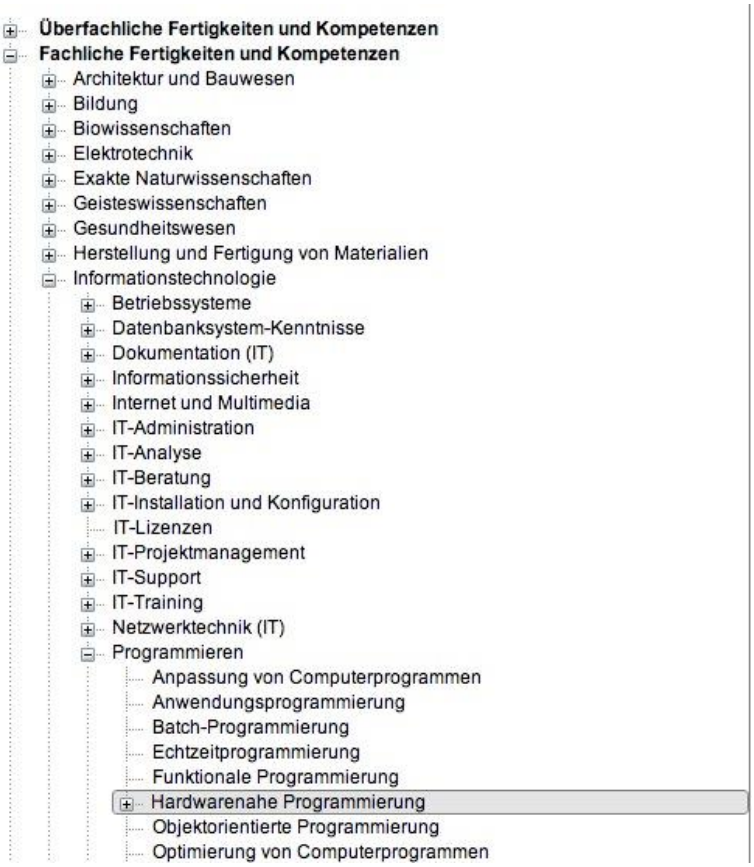

Abbildung 14: Darstellung des Eintrags für *Funktionale Programmierung* im fachspezifischen Teil des DISCO Thesaurus Explorers (Müller-Riedlhuber & Ziegler 2012b)

Bei der Integration des Klassifizierungsschemas haben die DISCO Autoren sowohl beim fachlichen als auch beim überfachlichen Bereich auf mehrere bestehende Standards zurückgegriffen. So kamen neben den Europäischen Kategorisierungsschemata *ISCED 1997*[37] *(International Standard Classification of Education)* und *ISCO-88*[38] *(International Standard Classification of Occupation)* mehrere nationale Standards wie etwa der *Deutsche Kompetenzkatalog*[39], das *Répertoire Opérationnel des Métiers et des Emplois*

37 http://www.uis.unesco.org/Education/Pages/international-standard-classification-of-education.aspx

38 http://laborsta.ilo.org/applv8/data/isco88e.html

39 http://download-portal.arbeitsagentur.de/files/registry.do?doNext=eulaAnzeigen

aus Frankreich oder die *Qualifikationsklassifikation*[40] des Österreichischen AMS zum Einsatz (Müller-Riedlhuber 2009).

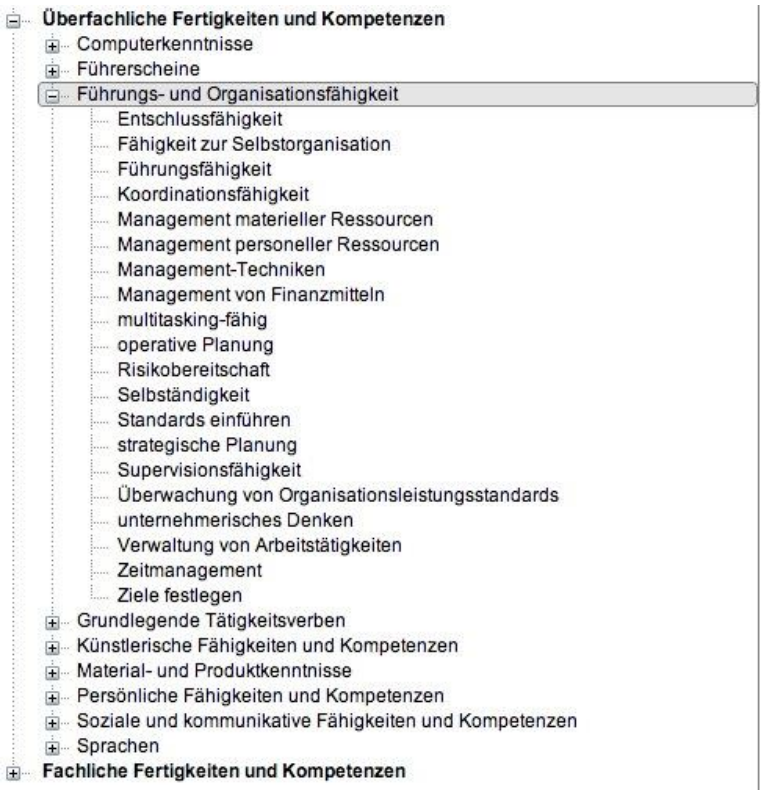

Abbildung 15: Darstellung aller neun Hauptkategorien aus dem Bereich *Überfachlichen Fertigkeiten und Kompetenzen* sowie aller Einträge für die Kategorie *Führungs- und Organisationsfähigkeit* (Müller-Riedlhuber & Ziegler 2012b)

Der DISCO Thesaurus wurde ursprünglich mit dem Ziel geschaffen, als unterstützende Ressource für die „internationale Vergleichbarkeit von Fertigkeiten und Kompetenzen in Anwendungen wie Lebensläufen, E-Portfolios, Stelleninseraten und Matching-Systemen sowie Beschreibungen von Qualifikationen und Lernergebnissen“ zu dienen. In Hinblick auf das oben angegebene Schema kann er zur Klasse der manuell erstellten Thesauri zum Einsatz in Informationssuchsystemen gezählt werden (Müller-Riedlhuber & Ziegler 2012).

40 http://www.forschungsnetzwerk.at/downloadpub/AMSinfo218.pdf

Das ROBUS Informationssuchsystem nutzt den DISCO Thesaurus und fügt dem ursprünglichen Anwendungsspektrum so ein zusätzliches Einsatzgebiet hinzu.
Die fachlichen Fertigkeits- und Kompetenzbegriffe des DISCO Thesaurus können somit genutzt werden, um eine Aussage darüber zu treffen, ob - beziehungsweise inwieweit - ein bestimmtes Wort relevant für einen bestimmten Mitarbeiter oder eine bestimmte Mitarbeiterin eines Unternehmens ist. Eine detaillierte Beschreibung der entsprechenden Methode des ROBUS Verfahrens findet sich in Kapitel 5.1.5.

4.4 Methoden der Computerlinguistik

4.4.1 Tokenisierung

Unter dem Vorgang der Tokenisierung versteht man in der Computerlinguistik laut Mikheev (Mikheev 2004) die Zerlegung eines elektronischen Textstromes in separate Einheiten – die sogenannten Tokens. Elektronische (auch: digitale) Textdaten werden in der Regel als Reihe von spezifischen Schriftzeichen (engl.: Character) repräsentiert, die im Rahmen des Tokenisierungsvorgangs in einzelne linguistische Elemente (Wörter, Nummern, Interpunktionszeichen, etc.) unterteilt werden. Dabei gilt es auch den eigentlichen textuellen Inhalt von gegebenenfalls inkludierten Steuer- und Formatierungszeichen (wie beispielsweise Informationen zu Schriftart, Absatzformatierung oder Zeilenumbrüchen) zu bereinigen. Dieser Schritt wird häufig vorab zur eigentlichen Tokenisierung in einem gesonderten Arbeitsvorgang - dem sogenannten Preprocessing (Vorverarbeitung) - durchgeführt.
Die Tokenisierung von elektronischen Texten gilt heute als unverzichtbare Grundvoraussetzung für alle Methoden der automatisierten Sprachverarbeitung. Als die elementarste

Vorgehensweise zur Identifikation von einzelnen Tokens beschreibt Mikheev die Strategie, den zu analysierenden Textstrom anhand von Leerzeichen und Interpunktionen zu segmentieren. Diese Strategie kann mit relativ einfachen Mitteln wie beispielsweise regulären Ausdrücken[41] (engl.: Regular Expression, Regex) umgesetzt werden (Mikheev 2004).

```
Another ex-Golden Stater, Paul Stankowski from Oxnard, is contending
for a berth on the U.S. Ryder Cup team after winning his first PGA Tour
event last year and staying within three strokes of the lead through
three rounds of last month's U.S. Open. H.J. Heinz Company said it
completed the sale of its Ore-Ida frozen-food business catering to the
service industry to McCain Foods Ltd. for about $500 million.
It's the first group action of its kind in Britain and one of
only a handful of lawsuits against tobacco companies outside the
U.S. A Paris lawyer last year sued France's Seita SA on behalf of
two cancer-stricken smokers. Japan Tobacco Inc. faces a suit from
five smokers who accuse the government-owned company of hooking
them on an addictive product.
```

Abbildung 16: Eingangstextstrom für die Tokenisierung (Manning et al. 2013)

Als Ergebnis des Tokenisierungsvorgangs liefern die meisten Systeme wiederum einen Textstrom, innerhalb dessen jeder vom System identifizierte Token durch ein Leerzeichen vom nächstfolgenden Token getrennt ist. Abbildung 16 zeigt einen Textstrom in englischer Sprache, der mithilfe des bekannten Stanford Tokenizers (Manning et al. 2013) analysiert werden soll. Das Ergebnis dieses Vorgangs ist in Abbildung 17 dargestellt. Daraus ist klar ersichtlich, dass neben Wörtern auch Interpunktionszeichen (wie beispielsweise Kommas) als separate Tokens identifiziert werden.

[41] http://www.lrz.de/services/schulung/unterlagen/regul/

```
Another ex-Golden Stater , Paul Stankowski from Oxnard , is contending
for a berth on the U.S. Ryder Cup team after winning his first PGA Tour
event last year and staying within three strokes of the lead through
three rounds of last month 's U.S. Open .
H.J. Heinz Company said it completed the sale of its Ore-Ida frozen-food
business catering to the service industry to McCain Foods Ltd. for about
$ 500 million .
It 's the first group action of its kind in Britain and one of only a
handful of lawsuits against tobacco companies outside the U.S. .
A Paris lawyer last year sued France 's Seita SA on behalf of two
cancer-stricken smokers .
Japan Tobacco Inc. faces a suit from five smokers who accuse the
government-owned company of hooking them on an addictive product .
```

Abbildung 17: Durch Leerzeichen voneinander getrennte Tokens als Ergebnis der Tokenisierung des Textes aus obiger Abbildung (Manning et al. 2013)

Dieser simple aber wirkungsvolle Ansatz der Identifikation von Tokengrenzen auf Basis von Leerzeichen und Interpunktionen wird heute als Grundlage für viele Systeme und deren weiterführende Verarbeitung herangezogen und liefert für zahlreiche Forscher ein zufriedenstellendes Ergebnis. Aufgrund dessen wird der Vorgang der Tokenisierung von vielen als triviale Aufgabe angesehen, der keine besondere Beachtung geschenkt zu werden braucht. Dass dies keineswegs der Fall ist, bekräftigt jedoch Mikheev nicht zuletzt in folgender Aussage: „Tokenization is usually considered as a relatively easy and uninteresting part of text processing for languages like English and other segmented languages where words are separated by blank and other punctuation. However, even in this languages there are cases where tokens are written with no explicit boundaries between them, and sometimes what seem to be two tokens in fact form one and vice versa. Ambiguous punctuation, hyphenated words, clitics, apostrophes, etc. largely contribute to the complexity of tokenization."

Aus obigem Zitat wird ersichtlich, dass eine „einfache" Tokenisierung auf Basis von Leerzeichen und Interpunktionen ausnahmslos für segmentierte Sprachen angewandt werden kann. Diese Art von Sprachen – dazu zählen etwa Deutsch, Englisch, Französisch und Spanisch – verfügen über explizite Merkmale (zum Beispiel Leerzeichen), um Wortgrenzen zu kennzeichnen. Im Gegensatz dazu existieren bei nicht-segmentierten Sprachen – dazu gehören beispielsweise Koreanisch, Japanisch, Thai und Chinesisch – keinerlei explizite Zeichen, die von der Methode zum Zwecke der Tokenisierung ausgewertet werden können. (SORNLERTLAM-VANICH et al. 2007)

Wie Mikheev im zweiten Teil des obigen Zitates ausführt, ist aber selbst bei segmentierten Sprachen eine pauschale Identifikation von Tokens anhand von Leerzeichen und Interpunktionen nicht ausreichend. Speziell im Bereich von Eigennahmen, Abkürzungen sowie Datums- und Währungsangaben führt dieses Vorgehen zu unzureichenden Ergebnissen. Verschärft wird diese Problematik noch zusätzlich durch den Umstand, dass der Tokenisierungsvorgang bei den meisten Systemen bereits in einer sehr frühen Verarbeitungsphase stattfindet und dass sich dadurch Fehler, die in dieser Phase entstehen, auf alle weiteren Arbeitsschritte auswirken. Festzuhalten ist daher, dass sich die Qualität der Tokenisierung maßgeblich auf die Qualität des Gesamtsystems auswirkt (Mikheev 2004).

4.4.2 Maximum-Entropie Tokenisierung mit OpenNLP

OpenNLP ist eine frei verfügbare Werkzeugsammlung zur Analyse und Verarbeitung von natürlichsprachlichen Texten. Sie wird unter dem Dach der Apache Software Foundation[42] seit 2004 entwickelt und als quelloffene Software (engl.: Open Source Software) zum Download[43] bereit gestellt. Aktuell (Stand: November 2013) ist die OpenNLP Bibliothek in der Version 1.5.3 verfügbar. Zu den wesentlichen Komponenten von OpenNLP zählen neben der Satzerkennung und der Tokenisierung die Erkennung von Wortarten (engl.: Part-of-Speech Tagging, POS-Tagging), die Erkennung von Eigennamen (engl.: Named Entity Recognition), die flache und die tiefe Satzanalyse (engl.: Chunking, Parsing) sowie die Koreferenzauflösung (engl.: Coreference Resolution). (Apache-OpenNLP-Development-Community 2013a)
Die OpenNLP Bibliothek stellt insgesamt drei verschiedene Tokenizer zur Verfügung:

[42] Apache Lizenz: http://www.apache.org/licenses/LICENSE-2.0

[43] OpenNLP Download: http://opennlp.apache.org/cgi-bin/download.cgi

- *Whitespace Tokenizer:* stellt die einfachste Form eines Tokenizers dar; es werden Leerzeichen (engl.: Whitespace) zur Identifikation der Tokengrenzen herangezogen
- *Simple Tokenizer:* dieser Tokenizer klassifiziert jedes einzelne Zeichen des zu analysierenden Textes und fasst alle zusammenhängenden Zeichen der selben Klasse zu einem Token zusammen
- *Learnable Tokenizer*: diese Implementierung enthält ein maschinelles Lernverfahren und ermittelt die Tokens beziehungsweise ihre Grenzen basierend auf der Maximum-Entropie-Methode

Alle drei Tokenizer werden sowohl als fertig ausführbare Programme (engl.: CLT - Command Line Tool) als auch in Form einer integrierbaren Programmierschnittstelle (engl.: API – Application Programming Interface) zur Verfügung gestellt. (Apache-OpenNLP-Development-Community 2013c)

```
Position Summary: We are looking for a U.S.-based Software Engineer/Web
Developer specialized in web interface and database programing. The
incumbent shall be a self-starter, highly organized should be able to
prioritize in order to meet deadlines (project management skills are +);
Position Responsibilities: Develop, manage and support interfaces
between our internal systems, our website and our partner websites
(APIs). Maintain and develop our website. Integrate various external
database systems with multiple other databases. Write server-side code
for web-based applications. Apply front-end technologies including i.e.
HTML5, CSS2+, JavaScript 1.8. Maintain our Linux and Windows Servers.
```

Abbildung 18: englischsprachiger Eingangstext zur exemplarischen Tokenisierung mit OpenNLP

Ähnlich wie der im vorigen Abschnitt beschriebene Stanford Tokenizer, kennzeichnen auch die OpenNLP Tokenizer die einzelnen Tokengrenzen durch das Einfügen eines Leerzeichens an der jeweiligen Position. Die nachfolgende Abbildung zeigt die Ausgabe eines Textstromes, der mit Hilfe des Simple Tokenizers analysiert wurde. Dabei ist zu

erkennen, dass das Programm nicht nur Leerzeichen, sondern auch Interpunktionen wie beispielsweise „.“, „-“, „/“, „;“ als Tokengrenzen interpretiert und dementsprechend jeweils ein Leerzeichen einfügt.

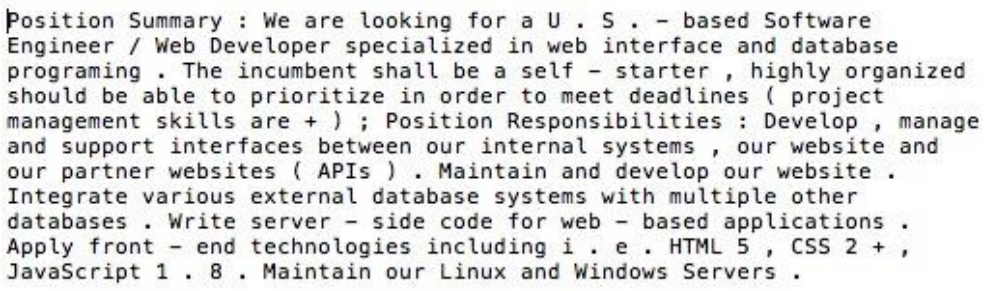

Abbildung 19: Ergebnis der Tokenisierung des Eingangstexts mittels des Simple Tokenizers von OpenNLP

Der gleiche Eingabetext führt beim Einsatz des Learnable Tokenizers zu einem unterschiedlichen Ergebnis. Wie der untenstehenden Abbildung entnommen werden kann, wird nicht jedes Interpunktionszeichen auch als Tokengrenze identifiziert. Im Gegensatz zum Simple Tokenizer wird beispielsweise die Zeichenkette „U.S.-based“ durch den Learnable Tokenizer nicht in sechs einzelne Tokens aufgeteilt. Vielmehr wird diese Zeichenkette als ein zusammengehöriger Ausdruck interpretiert und dementsprechend als ein einziger Token abgebildet.

Wie der Name bereits erahnen lässt, implementiert der Learnable Tokenizer ein maschinelles Lernverfahren. Dieses beruht auf der Maximum-Entropie-Methode und benötigt zur Durchführung des Tokenisierungsvorgangs ein zugrundeliegendes Maximum-Entropie-Modell mit den benötigten Eigenschaftsstatistiken (Berger et al. 1996).

```
Position Summary : We are looking for a U.S.-based Software Engineer/Web
Developer specialized in web interface and database programing . The
incumbent shall be a self-starter , highly organized should be able to
prioritize in order to meet deadlines ( project management skills are
+ ) ; Position Responsibilities : Develop , manage and support
interfaces between our internal systems , our website and our partner
websites ( APIs ) . Maintain and develop our website . Integrate various
external database systems with multiple other databases . Write server-
side code for web-based applications . Apply front-end technologies
including i .e . HTML5 , CSS2+ , JavaScript 1.8. Maintain our Linux and
Windows Servers .
```

Abbildung 20: Ergebnis der Tokenisierung des Eingangstexts mittels des Learnable Tokenizers von OpenNLP

OpenNLP stellt eine Reihe von vordefinierten Maximum-Entropie-Modellen für unterschiedliche Sprachen zur Verfügung. Für den Learnable Tokenizer können Modelle für Dänisch, Englisch, Deutsch, Niederländisch, Portugiesisch und Schwedisch von der Download-Seite[44] bezogen werden. Auch zum Trainieren der Modelle mittels unterschiedlicher Korpora stellt OpenNLP sowohl ein gebrauchsfertiges Programm als auch eine integrierbare Java-basierte Programmierschnittstelle zur Verfügung (Apache-OpenNLP-Development-Community 2013c).

4.4.3 Stemming

Wörter mit gleicher oder ähnlicher Bedeutung können in natürlichsprachlichen Texten in unterschiedlichen Formen beziehungsweise Ableitungen auftreten. So kann beispielsweise das deutsche Verb *gehen* je nach Person, Numerus und Tempus in den Flexionsformen *gehe, gehst, geht, gehen, ging, gingst, gingen, gingt* oder *gegangen* vorkommen. Auch derivationsmorphologisch können Wortvarianten wie z.B. *know, knowledge, knowable* abgeleitet werden.

Ziel der Stemming-Methode ist es, die Flexionsformen von Wörtern durch die Anwendung von regelbasierten Verfahren auf eine reduzierte Form – den Stamm (engl.: „stem“)

[44] OpenNLP Download-Seite für Maximum-Entropie-Modell: http://opennlp.sourceforge.net/models-1.5/

- zu reduzieren und dadurch die Ergebnisse von Informationssuchverfahren zu verbessern (Manning & Schuetze 1999).

In der Literatur existiert heute eine Vielzahl von Stemming-Algorithmen und Implementierungen davon. Der erste Stemmer, der speziell für die Informationssuche entwickelt wurde, geht zurück auf den Algorithmus von (Lovins 1968). Dieser basiert auf einem Katalog von allgemeinen Suffixen (zum Beispiel: „-SES“, „-TION“, „-ING“) und einem darauf aufbauenden Regelwerk. Die grundlegende Vorgehensweise dieses Algorithmus besteht darin, das rechte Ende jedes zu stemmenden Wortes auf das Vorhandensein eines Suffixes aus dem vordefinierten Katalog zu untersuchen und – sofern vorhanden – den Suffix zu entfernen.

```
private final void step3() { if (k == 0) return; /* For Bug 1 */ switch (b[k-1])
{
    case 'a': if (ends("ational")) { r("ate"); break; }
        if (ends("tional")) { r("tion"); break; }
        break;
    case 'c': if (ends("enci")) { r("ence"); break; }
        if (ends("anci")) { r("ance"); break; }
        break;
    case 'e': if (ends("izer")) { r("ize"); break; }
        break;
    case 'l': if (ends("bli")) { r("ble"); break; }
        if (ends("alli")) { r("al"); break; }
        if (ends("entli")) { r("ent"); break; }
        if (ends("eli")) { r("e"); break; }
        if (ends("ousli")) { r("ous"); break; }
        break;
    case 'o': if (ends("ization")) { r("ize"); break; }
        if (ends("ation")) { r("ate"); break; }
        if (ends("ator")) { r("ate"); break; }
        break;
    case 's': if (ends("alism")) { r("al"); break; }
        if (ends("iveness")) { r("ive"); break; }
        if (ends("fulness")) { r("ful"); break; }
        if (ends("ousness")) { r("ous"); break; }
        break;
    case 't': if (ends("aliti")) { r("al"); break; }
        if (ends("iviti")) { r("ive"); break; }
        if (ends("biliti")) { r("ble"); break; }
        break;
    case 'g': if (ends("logi")) { r("log"); break; }
} }
```

Abbildung 21: Auszug aus der Porter Stemmer Implementierung nach (Porter 2005)

Der wohl am bekannteste Algorithmus für englischsprachige Texte ist der sogenannte „Porter Stemming Algorithmus“, der erstmals 1980 von Martin F. Porter veröffentlicht wurde (Porter 1980). Er zeichnet sich im Vergleich zum Lovins Algorithmus durch eine

deutlich reduzierte Komplexität aus und unterscheidet sich von diesem in zwei wesentlichen Aspekten: einerseits verwendet der Porter Algorithmus deutlich weniger Regeln hinsichtlich der Suffix-Entfernung (60 anstatt 294) und andererseits kennt er - im Gegensatz zu Lovins – keine kontext-sensitiven Heuristiken im Hinblick auf die Mindestlänge des verbleibenden Stems, sondern definiert stattdessen eine konstante Größe von 2. Durch das iterative Vorgehen in fünf Phasen erreicht der Porter Algorithmus trotz seiner verminderten Komplexität äußerst effektive Ergebnisse und kommt auch heute noch in vielen Einsatzgebieten zur Anwendung (Willett 2006).
Abbildung 21 zeigt den Auszug einer Java-Implementierung, die Porter selbst entwickelt und veröffentlich hat (Porter 2005). Für eine vollständige Erläuterung des Algorithmus sowie verfügbarer Implementierungen sei an dieser Stelle auf die umfangreiche bestehende Literatur, beispielsweise (Porter 1980) oder (Willett 2006), verwiesen.

```
This position would be ideal for an adaptable, experi-
enced and highly skilled candidate with a sound under-
standing of C#, .Net and OOP, and a thorough understand-
ing of writing secure code. Using your in depth experi-
ence with RESTful and SOAP web services, you will drive
the adoption of new technologies and techniques, spe-
cialising in .Net and increasing your Java knowledge.
As Senior Web Developer, you will also ideally have
knowledge of ASP.Net MVC, Test Driven Development, ORM
Frameworks, Java frameworks (such as Spring) and front-
end JavaScript experience. You will have the chance to
work on numerous exciting projects for the client's cus-
tomers; taking the lead when building web applications
and advising the rest of the team. You will be passion-
ate, innovative and keen to succeed! If this sounds like
you - apply today!
```

Abbildung 22: englischsprachiger Eingangstext zum exemplarischen Stemming mittels Porter Stemmer

Der englischsprachige Text einer Stellenausschreibung (Auszug)[45] in Abbildung 22 wurde mittels der Java

[45] Quelle: http://jobview.monster.co.uk/Senior-Web-Developer-C-Net-Java-OOP-MVC-RESTful-Job-Sheffield-Yorkshire-UK-132067247.aspx; zugegriffen am 30.03.2014

Implementierung von (Porter 2005) einem Stemming unterzogen. Das Ergebnis ist in Abbildung 23 dargestellt. Daraus ist die Arbeitsweise des Stemmers klar erkennbar. So wird beispielsweise das Wort *position* auf den Stem *posit*, das Wort *adaptable* auf den Stem *adapt*, usw. reduziert.

```
thi posit would be ideal for an adapt, experienc and
highli skill candid with a sound understand of c#, .net
and oop, and a thorough understand of write secur code.
us your in depth experi with rest and soap web servic,
you will drive the adopt of new technolog and techniqu,
specialis in .net and increas your java knowledg. as
senior web develop, you will also ideal have knowledg
of asp.net mvc, test driven develop, orm framework, java
framework (such as spring) and front-end javascript ex-
peri. you will have the chanc to work on numer excit
project for the client's custom; take the lead when
build web applic and advis the rest of the team. you
will be passion, innov and keen to succe! if thi sound
like you - appli todai!
```

Abbildung 23: Ergebnis des Stemming-Vorgangs des Eingangstexts mittels des Porter Stemmers

Durch diese Reduktion von mehreren Wortformen auf einen Stem lässt sich die Sensitivität (engl: Recall) von Informationssuchsystemen unter bestimmten Bedingungen steigern. Der Einsatz von Stemmern im Bereich der Informationssuche ist jedoch nicht unumstritten: Mehrere Untersuchungen haben gezeigt, dass Stemming zu keiner - beziehungsweise nur zu einer unwesentlichen - Verbesserung von Suchergebnissen im Allgemeinen führt. Vielmehr konnte belegt werden, dass Stemming zwar bei manchen Suchanfragen zu einer Verbesserung führt, aber bei anderen Suchanfragen sogar eine Verschlechterung verursachen kann (Manning & Schuetze 1999), (Hull 1996). Als Erklärung dafür nennen die zuvor genannten Autoren drei wesentliche Gründe:

(1) Die Reduktion von unterschiedlichen Wortformen auf einen gemeinsamen Stem führt unweigerlich zu Informationsverlusten, die sich negativ auf das Suchergebnis auswirken können. So reduziert beispielsweise der Porter Stemmer die Wörter

business (Geschäft) und *busy* (beschäftigt) auf den gleichen Stem *busi*, obwohl sie semantisch unterschiedliche Konzepte repräsentieren.

(2) In vielen Verfahren aus dem Bereich der natürlichen Sprachverarbeitung wurde gezeigt, dass Verbesserungen erzielt werden können, indem verwandte Wörter zu zusammenhängenden Einheiten (engl.: Multiwords, Chunks) gruppiert werden. Morphologische Worteigenschaften - wichtige Grundlagen für diese Verfahren - gehen jedoch im Zuge des Stemmings verloren.

(3) Der Großteil der untersuchten Texte existiert in englischer Form und verfügt daher über wenig Morphologie, was wiederum den Nutzen von Stemmern von vornherein einschränkt.

4.4.4 Lemmatisierung

Unter Lemmatisierung versteht man im Bereich der natürlichen Sprachverarbeitung eine morphologische Analyse zur Bestimmung des Lemmas (Grundform, auch: Lexem) eines Wortes. Im Unterschied zum Stemming (siehe Kapitel 4.4.3) geschieht dies jedoch nicht ausschließlich durch ein regelbasiertes Ersetzen von Suffixen, sondern durch eine computerlinguistische Analyse, bei der wesentliche grammatikalische Eigenschaften, wie beispielsweise die Wortart (engl. Part-of-Speech), bestimmt werden. Lemmatisierungsprogramme liefern im Allgemeinen bessere Ergebnisse im Vergleich zu Stemmern, sind jedoch komplexer in der Umsetzung und bedürfen mehr Ressourcen bei der Verarbeitung (Manning et al. 2008).

```
this position will be ideal for a adaptable , experience
and high skill candidate with a sound understanding of
C# , . Net and oop , and a thorough understanding of
write secure code . use your inn depth experience with
restful and soap web service , you will drive the adop-
tion of new technology and technique , specialise in .
Net and increase your Java knowledge . as senior web
developer , you will also ideal have knowledge of asp .
Net muc , test drive development , orm frameworks , Java
framework ( such as spring ) and front-end javascript
experience . you will have the chance to work on numerous
exciting project for the client customer ; take the lead
when build web application and advise the rest of the
team . you will be passionate , innovative and keen to
succeed ! if this sound like you - apply today
```

Abbildung 24: Ergebnis des Lemmatisierungsvorgangs des Eingangstexts aus Abbildung 22 mittels des Tools MorphAdorner[46]

Die Abbildung oben zeigt das Ergebnis des Lemmatisierungsvorgangs des Textbeispiels aus Abbildung 22. Die Lemmatisierung wurde mit Hilfe des frei verfügbaren Tools „MorphAdorner V2.0“ durchgeführt (Northwestern University Information Technology 2013). Aus dem Beispiel wird ersichtlich, dass der Lemmatisierer – im Gegensatz zum oben gezeigten Stemming-Verfahren - nicht auf dem reinen Ersetzen von Zeichenketten basiert, sondern die lexikalische Grundform jedes Wortes bestimmt. So wird beispielsweise *position* nicht auf den Stem *posit* reduziert, sondern bleibt als *position* im Text erhalten, während *would* durch seine Grundform *will* ersetzt wird.

Trotz der besseren Ergebnisqualität von Lemmatisierungsprogrammen erhalten im Bereich von Informationssuchsystemen häufig Stemmer den Vorzug. Als Grund dafür nennen (Perera & Witte 2005) neben dem größeren Ressourcenbedarf bei der Verarbeitung vor allem den hohen Kosten- beziehungsweise Zeitaufwand zur Erlangung eines adäquaten Lexikons. Im Gegensatz zu Stemmern können Lemmatisierer nicht mit reinen Heuristiken arbeiten, sondern benötigen zwingend ein Lexikon, das alle Wörter und deren Flexionsformen (Vollformlexikon, engl.: Full-form Lexicon) oder alle Lemma und ein zugehöriges Regelwerk zur Ableitung aller Flexionsformen (Vollformlexikon, engl.:

46 Quelle: (Northwestern University Information Technology 2013)

Base-form Lexicon) enthält. Die manuelle Erstellung solcher Lexika ist jedoch mit enormem Aufwand verbunden. Auch die Nutzung von bestehenden kommerziell verfügbaren Standard-Lexika ist nur bedingt zielführend, da domänenspezifische Wörter nicht berücksichtigt werden. Als Lösungsansatz präsentieren die Autoren einen selbstlernenden kontextsensitiven Lemmatisierer, der in der Lage ist, das benötigte Lexikon aus den zu analysierenden Dokumenten automatisiert zu generieren (Perera & Witte 2005).

4.4.5 Auswirkungen von Stemming und Lemmatisierung

Die Darstellung in Abbildung 25 zeigt das Ergebnis einer exemplarischen Untersuchung einer Stellenausschreibung[47]. Im ersten Schritt wurden die Wörter im Ausschreibungstext mit Hilfe des Porter-Stemmers aus Kapitel 4.4.3 reduziert (Absatz 2, „GESTEMMTER TEXT“). Die durch den Stemmer gelöschten Affixe sind in der Abbildung durch Unterstriche („___“) gekennzeichnet. Die gelb hinterlegten Zeichenketten stellen Fehler dar, bei denen der Stemming-Algorithmus unterschiedliche semantische Konzepte auf einen gemeinsamen Stem reduziert hat („Overstemming“). So wurden z.B. *RESTful* durch *rest* und *customers* durch *custom* ersetzt.
Die rot hinterlegten Stellen dagegen kennzeichnen Passagen, bei denen domänenspezifische Konzepte des Originaltextes durch den Stemmer verändert wurden. So wurde beispielsweise der Fachbegriff *Test Driven Development* auf *test driven develop* reduziert. Diese spezielle Form des Overstemmings tritt bei domänenspezifischen Begriffen und Eigennamen auf, da deren spezielle Muster nicht durch die standardmäßigen – domänenunabhängigen – Heuristiken von Stemming-Algorithmen abgebildet werden (Willett 2006).

[47] Quelle: http://jobview.monster.co.uk/Senior-Web-Developer-C-Net-Java-OOP-MVC-RESTful-Job-Sheffield-Yorkshire-UK-132067247.aspx; zugegriffen am 30.03.2014

Wie im Eingangskapitel dieser Arbeit erläutert, ist es das Ziel des ROBUS Verfahrens, Stellenausschreibungstexte zu durchsuchen und daraus repräsentative Rollenprofilvektoren zu generieren. Stellenausschreibungstexte weisen jedoch einen überdurchschnittlich hohen Anteil an domänenspezifischen Begriffen und Eigennamen auf. Der oben beschriebene Fehler hätte dementsprechend einen höheren negativen Einfluss auf das System. Auf Grund des von vornherein als eher gering anzunehmenden Nutzens eines Stemmers (siehe Kapitel 4.4.3) sowie des hier beschriebenen Fehlers kommt Stemming im ROBUBS System somit nicht zum Einsatz.

ORIGINALTEXT

This position would be ideal for an adaptable, experienced and highly skilled candidate with a sound understanding of C#, .Net and OOP, and a thorough understanding of writing secure code. Using your in depth experience with RESTful and SOAP web services, you will drive the adoption of new technologies and techniques, specialising in .Net and increasing your Java knowledge. As Senior Web Developer, you will also ideally have knowledge of ASP.Net MVC, Test Driven Development, ORM Frameworks, Java frameworks (such as Spring) and front-end JavaScript experience. You will have the chance to work on numerous exciting projects for the client's customers; taking the lead when building web applications and advising the rest of the team. You will be passionate, innovative and keen to succeed! If this sounds like you - apply today!

GESTEMMTER TEXT

Thi_ posit___ would be ideal for an adapt____, experience_ and highli skill__ candid___ with a sound understand___ of c#, .net and oop, and a thorough understand___ of write__ secur_ code. Us___ your in depth experi____ with rest___ and soap web service_, you will drive the adopt___ of new technolog___ and technique_, specialis___ in .net and increase__ your java knowledge_. as senior web develop__, you will also ideal__ have knowledge_ of asp.net mvc, test driven develop____, orm framework_, java framework_ (such as spring) and front-end javascript experi____. you will have the chanc_ to work on numer___ excit___ project_ for the client's custom___; take__ the lead when build___ web applic_____ and advis___ the rest of the team. you will be passion___, innov_____ and keen to succe__! if thi_ sound like you - appli todai!

LEMMATISIERTER TEXT

this position will be ideal for a_ adaptable , experience_ and high__ skill__ candidate with a sound understanding of C# , . Net and oop , and a thorough understanding of write__ secure code . use__ your inn depth experience with restful and soap web service_, you will drive the adoption of new technology__ and technique_, specialise_ in . Net and increase__ your Java knowledge . as senior web developer , you will also ideal__ have knowledge of asp . Net muc , test drive_ development , orm frameworks , Java framework_ (such as spring) and front-end javascript experience . you will have the chance to work on numerous exciting project_ for the client_ customer ; take__ the lead when build__ web application_ and advise__ the rest of the team . you will be passionate , innovative and keen to succeed ! if this sound like you - apply today

Abbildung 25: Unterschiede und Fehler beim Stemming und Lemmatisieren von Stellenausschreibungstexten

Im Gegensatz zum gestemmten Text weist der lemmatisierte Text in Abbildung 25 keine Fehler durch „Overstemming“ auf. Alle Wörter werden durch das Lemmatisierungsverfahren auf ihre korrekte Grundform zurückgeführt. Jedoch zeigen sich an den rot markierten Stellen ebenfalls Probleme im Zusammenhang mit domänenspezifischen Begriffen, Eigennamen und Abkürzungen. So wird beispielsweise die Technologiebezeichnung *ASP.Net* in mehrere Einzeltokens (*asp* . *net*) aufgeteilt. Die Abkürzung *MVC* (Model View Controller) hingegen wird durch den Lemmatisierer in *muc* umgewandelt. Diese Fehler wirken sich auf jede nachfolgende Komponente in der Bearbeitungskette aus und würden somit auch bei ROBUS zu unerwünschten Ergebnissen führen (vgl. Kapitel 5.1). Um dieser Problematik zu entgegnen, müsste für den Lemmatisierungsvorgang ein eigenständiges, domänenspezifisches Lexikon entwickelt werden (Perera & Witte 2005). Das Entwickeln bzw. Generieren eines solchen Lexikons ist jedoch ein langfristiges Vorhaben und wurde daher im Rahmen dieser Arbeit nicht durchgeführt. Für die Evaluation der gegenständlichen Arbeit hat dies jedoch keine Auswirkungen, da die Lemmatisierung weder im Vergleichs- („Baseline“) noch im Testsystem (ROBUS) zum Einsatz kommt (vgl. Kapitel 6.1).

5 Rollenbasierte Unternehmenssuche mit ROBUS

Das ROBUS Verfahren wurde mit dem Ziel entwickelt, die Suche in unstrukturierten Daten innerhalb eines Unternehmens zu verbessern, indem zusätzlich zur textuellen Suchanfrage kontext-bezogene Informationen über suchende Benutzer/innen einbezogen werden. Jede/r Benutzer/in einer Unternehmenssuchmaschine nimmt als Mitarbeiter/in eine bestimmte Rolle im Unternehmen ein. Beispiele für gängige Unternehmensrollen sind „Web Entwickler/in", „Marketingleiter/in" oder „Netzwerktechniker/in". Die konkrete Rolle, die ein/e Mitarbeiter/in im Unternehmen einnimmt, ist ein wesentlicher Bestandteil des Benutzerkontexts und reflektiert dessen langfristige Informationsbedürfnisse. So erwartet sich ein/e Benutzer/in mit der Rolle „Web Entwickler/in" von einer Unternehmenssuchmaschine andere Suchergebnisse, als ein/e Benutzer/in mit der Rolle „Marketingleiter/in", auch wenn beide die gleichen Suchbegriffe angeben. Genau diesen Sachverhalt nutzt ROBUS, um anhand von Rollenprofilen, die jedem Benutzer zugeordnet sind, Suchergebnisse an die langfristigen Informationsbedürfnisse der Benutzer anzupassen (Reichhold et al. 2011).

Eine wesentliche Grundlage für den Einsatz von ROBUS ist, dass für jede/n Benutzer/in eine eindeutige Rollenzuordnung verfügbar ist. Diese Rolleninformation kann entweder explizit gepflegt (durch manuelles Zuordnen einer Unternehmensrolle zu einem Benutzer) oder implizit aus vorhanden Metainformationen (z.B. Job-Titel, Stellenbeschreibungen, Berechtigungssystemen, u.ä.) abgeleitet werden.

Ausgehend von der Rollenbezeichnung (z.B. „Web Developer/in") erstellt ROBUS für jede vorhandene Rolle ein Rollenprofil. Diese Rollenprofile dienen dazu, eine Relation

zwischen jeder einzelnen Unternehmensrolle und den zu durchsuchenden textuellen Dokumenten schaffen zu können, und werden als gewichtete Termvektoren repräsentiert. So kann jede Unternehmensrolle durch einen Termvektor beschrieben werden, indem dieser eine bestimmte Anzahl von Termen (Wörtern) enthält, die für die jeweilige Rolle relevant sind. Zusätzlich enthält jeder Vektor *V* für jeden Term *t* genau ein Gewicht *w*, welches das Maß der Relevanz beschreibt.

$V_{WebDeveloper}\ (t, w) =$

(web, 3.05; developer, 2.56; asp, 1.83; php, 1.74; javascript, 1.56; ...)

Abbildung 26: Beispiel für einen gewichteten Termvektor der Rolle „Web Developer“

Um für jede Rolle genau jene Wörter zu identifizieren, die diese am zielführendsten beschreiben, analysiert ROBUS textuelle Stellenausschreibungen. Dabei macht sich das System den semantischen Zusammenhang zwischen Unternehmensrollen (z.B. „Web Developer“) und Stellenausschreibungen (siehe Abbildung 27) zu Nutze: Stellenausschreibungen beziehen sich immer auf konkrete Unternehmensrollen und enthalten die relevanten Terme, die zur Erstellung der Rollenprofile benötigt werden (vgl. die hervorgehobenen Begriffe in Abbildung 27) (Reichhold et al. 2012). Die Herausforderung für das System besteht nun darin, automatisiert die relevanten Terme von den nicht relevanten zu separieren und entsprechend ihrer Relevanz zu gewichten. ROBUS bedient sich dazu einer Kombination verschiedener computerlinguistischer Methoden, dem DISCO-Thesaurus sowie einer speziellen Gewichtungsfunktion, die im Folgenden näher beschrieben werden.

```
<job>
  <id>2931333</id>
  <position>
    <title>Software Engineer, Web Developer – Java, PHP, SQL, Front-End Technologies</title>
  </position>

  <description>Position Summary: We are looking for a Software Engineer / Web Developer
specialized in web interface and database programing. The incumbent shall be a self-
starter, highly organized should be able to  prioritize in order to meet deadlines (project
management skills are plus) Position Responsibilities: Develop, manage and support
interfaces between our internal systems, our website and our partner websites (APIs).
Maintain and develop our website. Integrate various external database systems with
multiple other databases. Write server-side code for web-based applications. Apply
front-end technologies including i.e. HTML, CSS, JavaScript. Maintain our Linux and
Windows Servers.
</description>

<skills-and-experience> 4+ years related work experience. Strong experience in interface
programming (APIs). Proficient in Java, PHP, XML, HTML, CSS, JavaScript, JQuery,
ASP.Net, ASP Classic, MySQL. In depth SQL development experience. Experience in proper
web programming techniques, best practices, software application standards etc.
Experience with the Linux and Windows server environment (web hosting and hosting
technologies) is a big plus. Computer proficiency with the ability to navigate through
several computer applications....
</skills-and-experience>
</job>
```

Abbildung 27: Auszug einer Stellenausschreibung aus dem Online-Portal linekdin.com

5.1 Generierung von Rollenprofilen

Wie in den vorherigen Kapiteln geschildert, sind Rollenprofile die grundlegende Voraussetzung für das Funktionieren der kontext-sensitiven Suche in ROBUS. Die Erstellung der den Profilen zugrunde liegenden Termvektoren ist daher von entscheidender Bedeutung für die Leistung des Suchalgorithmus. In (Reichhold et al. 2011) werden drei verschiedene Ansätze zur Erstellung solcher Vektoren vorgestellt:

(1) Relevante Terme und ihre Gewichtungen werden von Domänenexperten für jede Organisation manuell verwaltet und gepflegt.

(2) Relevante Terme werden durch die Mitarbeiter/innen selbst gepflegt, in dem diese auf persönlichen Profilseiten Schlagwörter eintragen. Diese Schlagwörter können dann systematisch ausgewertet und der entsprechenden Rolle zugeordnet werden.

(3) Relevante Terme werden unter Zuhilfenahme von unternehmensinternen Wissensdatenbanken („Enterprise Wikis“) aus den persönlichen Suchanfragen der Mitarbeiter/innen extrahiert.

Trotz gewisser Vorzüge, bestehen für jeden der oben beschriebenen Ansätze Nachteile, die einen Einsatz in ROBUS unmöglich machen (Reichhold et al. 2012): Ansatz (1) bedingt die Existenz von Personen, die im Unternehmen zur Verfügung stehen und über das nötige (unternehmensinterne) Wissen verfügen, um die relevanten Terme und deren Gewichtungen für alle existierenden Rollen zu bestimmen. Die manuelle Erstellung von Rollenprofilen ist nicht nur sehr zeit- und ressourcenintensiv, sondern darüber hinaus unflexibel und abhängig von den Entscheidungen einzelner Personen. Gegen die Ansätze (2) und (3) spricht, dass in beiden Fällen zusätzliche Systemkomponenten (Profilverwaltungssystem, Wissensdatenbanken) benötigt werden. Die Verfügbarkeit der benötigten Komponenten kann jedoch nicht vorausgesetzt werden und eine Neueinführung im Zuge von ROBUS verursacht einen erheblichen Zusatzaufwand für jede Organisation.
Darüber hinaus liegen die Systemkomponenten außerhalb des Einflussbereichs von ROBUS, haben jedoch massive Auswirkungen auf die Erstellung der Rollenprofile und damit auf die Suche selbst. Ein weiterer Nachteil von (2) und (3) liegt darin, dass auf personenbezogene Daten der Mitarbeiter/innen (Mitarbeiterprofil bzw. Historie der persönlichen Suchanfragen) zugegriffen wird. Dieser Umstand unterliegt in vielen Ländern bzw. Organisationen besonderen rechtlichen Bestimmungen (vgl. (Inneren 2010; Jaspers 2012)) und führt unter den Betroffenen oft zu großer Skepsis oder sogar Ablehnung gegenüber einem System. Des Weiteren beinhaltet jede der drei beschriebenen Varianten gewisse Tätigkeiten, die ausschließlich manuell durchgeführt werden können, was dem für ROBUS geforderten Ziel einer vollautomatischen Lösung widerspricht.
Daher wird in (Reichhold et al. 2012) eine völlig neue Vorgehensweise zur Erstellung von Termvektoren präsentiert, die keine der oben angeführten Einschränkungen aufweist und somit zur automatisierten Erstellung von Rollenprofilen für ROBUS verwendet werden kann. Ausgangspunkt für die Termextraktion sind dabei textuelle Stellenausschreibungen. Diese können sowohl von unternehmensinternen als auch externen Quellen be-

zogen werden. Die relevanten Stellenausschreibungen werden sodann einer computerlinguistischen Analyse unterzogen, die mit Hilfe verschiedener Werkzeuge (Satzerkennung, Tokenizer, POS Tagger) Termkandidaten identifiziert. Im nächsten Schritt werden aus den Termkandidaten mittels des standardisierten DISCO Thesaurus und einer speziellen Gewichtungsfunktion die Term-Gewicht-Tupel extrahiert und die entsprechenden Rollenprofile in Form von gewichteten Termvektoren erstellt. Die maßgeblichen Parameter für diese Funktion sind dabei

(1) die Existenz bzw. Position des Termkandidatens im DISCO Thesaurus

(2) die Vorkommenshäufigkeit des Termkandidatens innerhalb der Stellenausschreibung (Termfrequenz)

(3) die Vorkommenshäufigkeit des Termkandidatens in allen Stellenausschreibungen (Inverse Dokumenthäufigkeit)

(4) Die Vorkommensposition des Termkandidaten innerhalb der Stellenausschreibung

Auf Basis dieser Rollenprofile kann ROBUS die Relevanz für jedes verfügbare Dokument innerhalb der Organisation in Bezug auf eine bestimmte Unternehmensrolle ermitteln und bei der kontext-sensitiven Suche berücksichtigen, indem Suchergebnisse mit größerer Rollenrelevanz höher gereiht werden. Abbildung 28 zeigt einen schematischen Blick auf die automatisierte Profilerstellung in ROBUS. Die oben angeführten Komponenten bzw. Arbeitsschritte werden in den folgenden Abschnitten detailliert erläutert.

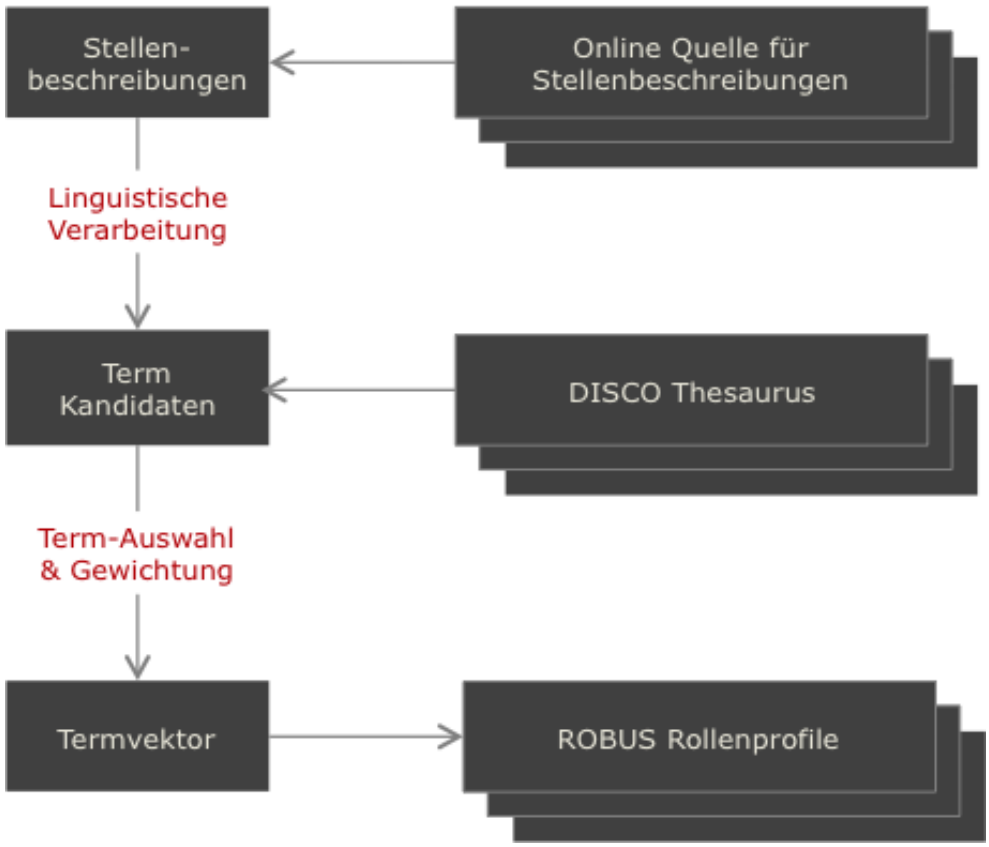

Abbildung 28: Schematischer Überblick der automatisierten Profilerstellung in ROBUS

5.1.1 Stellenausschreibungsdaten

Jede/r Mitarbeiter/in eines Unternehmens übt im Zuge der Arbeitstätigkeit bestimmte Funktionen aus und ist explizit oder implizit bestimmten Rollen innerhalb des Unternehmens zugeordnet. Jede Funktion wird einerseits durch die Anforderungen des Unternehmens und andererseits durch die Qualifikationen des ausübenden Mitarbeiters bestimmt. Unternehmen verfassen und veröffentlichen Stellenausschreibungen, um geeignete Mitarbeiter/innen für eine bestimmte Funktion zu finden. Eine Stellenausschreibung ist eine besondere Form eines textuellen Dokuments, das eine bestimmte Funktion (Stelle) innerhalb einer Organisation (Unternehmen) detailliert spezifiziert. Sie enthält Ausdrücke und

Formulierungen, die die gewünschten Fähigkeiten, Kompetenzen und Anforderungen in Bezug auf die beschriebene Funktion charakterisieren.
Des Weiteren finden sich in einer Stellenausschreibung in hohem Maße Fachbegriffe, die gezielt auf branchen- bzw. funktionsspezifische Aspekte (Technologien, Systeme, Anwendungen, etc.) verweisen (Loth et al. 2010). Ziel des Unternehmens ist es, mit Hilfe einer Stellenausschreibung jene Mitarbeiter/innen zu finden, deren Qualifikationen (Fähigkeiten, Fachwissen, Ausbildung, etc.) die größte Übereinstimmung mit den Anforderungen der Stelle aufweisen und die in Folge dessen die gewünschten Unternehmensrollen bekleiden können. Somit können Stellenausschreibungen als textuelle Beschreibungen von Unternehmensrollen interpretiert werden, die charakteristische Stichwörter und Formulierungen enthalten, die wiederum als langfristiger Benutzerkontext angesehen werden können und Aufschluss über die Informationsbedürfnisse jener Mitarbeiter/innen widerspiegeln, die der jeweiligen Rolle zugeordnet sind.

5.1.2 LinkedIn als Datenquelle für Stellenausschreibungen

Wie im vorherigen Abschnitt erläutert, analysiert ROBUS textuelle Stellenausschreibungen, um daraus automatisiert Rollenprofile zu generieren. Die dafür benötigten Ausschreibungstexte können aus verschiedenen Quellen, wie zum Beispiel Online-Jobportalen oder unternehmensinternen Personalinformationssystemen stammen. Das im Rahmen dieser Arbeit implementierte Evaluationssystem bezieht die Ausschreibungsdaten von der sozialen Netzwerkplattform „LinkedIn".
LinkedIn ist ein öffentlich zugänglicher Online-Dienst, der im Internet[48] erreichbar ist. Er richtet sich hauptsächlich an berufsorientierte Benutzer/innen, die mit Hilfe dieser Plattform ein persönliches Profil erstellen und detaillierte Angaben zu ihrem Lebenslauf sowie ihren (beruflichen) Interessen hinterlegen können. Des Weiteren erlaubt LinkedIn seinen Benutzer/inne/n sich mit anderen Mitgliedern zu verbinden („Benutzer zu meinem

[48] www.linekdin.com

Netzwerk hinzufügen“) und ermöglicht so die Entstehung von themenbezogenen Netzwerkgruppen. Mit 500 Millionen Seitenaufrufen pro Monat ist LinkedIn hinter Facebook[49] das zweitgrößte soziale Netzwerk weltweit (Papacharissi 2009).
Darüber hinaus bietet LinkedIn aber auch einen eigenen Stellenmarkt (Jobbörse) mit umfangreichen Möglichkeiten für Arbeitgeber und Arbeitssuchende. Unternehmen können auf diese Weise Stellenausschreibungen veröffentlichen während interessierte Personen gezielt nach vakanten Stellen suchen können. Die Eingabe von Stellenausschreibungsdaten erfolgt nach einem von LinkedIn vorgegebenen Formular (siehe Abbildung 29), das definierte Felder für die Eingabe von Texten und Metadaten bereit stellt; auf der Plattform veröffentlichte Ausschreibungstexte können somit als teil-strukturierte Daten betrachtet werden.

[49] www.facebook.com

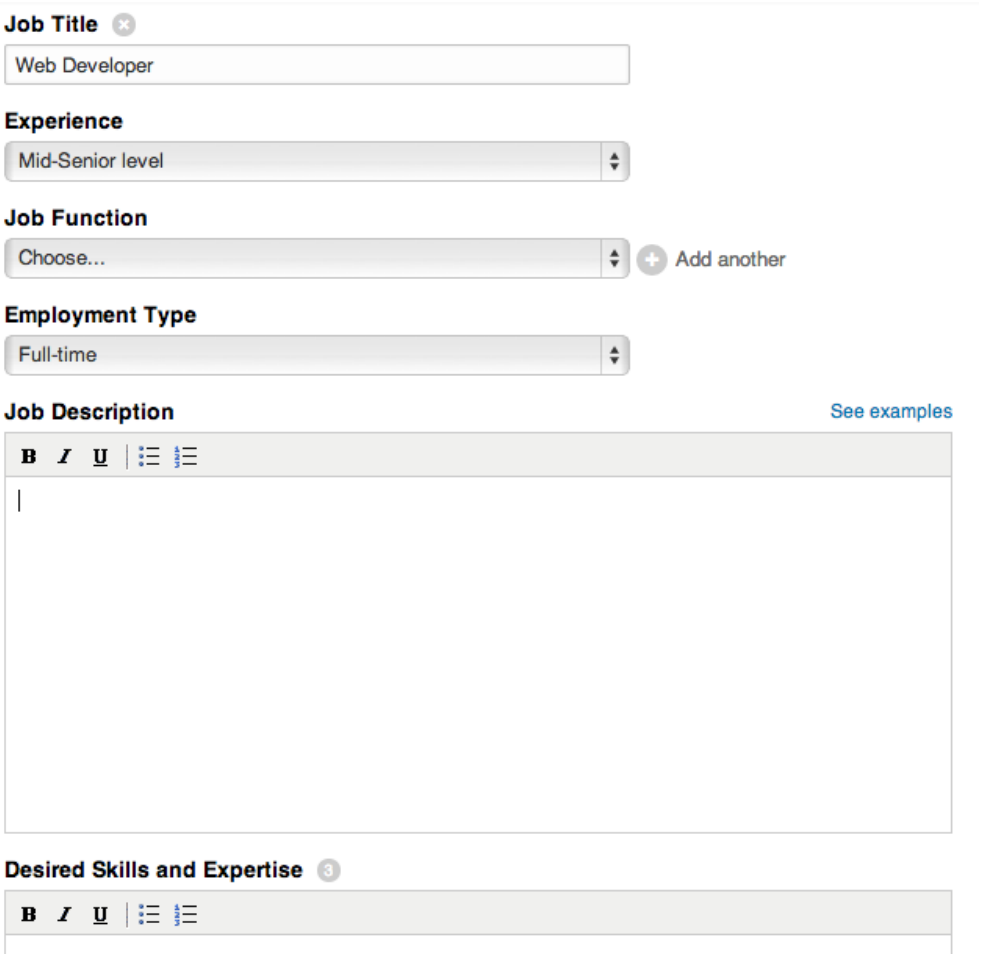

Abbildung 29: Screenshot der Eingabemaske für Stellenausschreibungen (Auszug); Quelle: www.linkedin.com, 25.03.2013

Zu den für ROBUS relevanten Feldern zählen neben dem eindeutigen Kennzeichner („ID“) noch der Titel („Job Title“), die allgemeine Stellenbeschreibung („Description“) sowie der Abschnitt „Skills and Experience“, der speziell für Angaben zu den geforderten Fähigkeiten und Berufserfahrungen vorgesehen ist.

5.1.3 Selektion von relevanten Stellenausschreibungen

Aus der Menge aller vorhandenen bzw. verfügbaren Ausschreibungstexte sollen für die automatisierte Profilerstellung in ROBUS nur jene in Betracht gezogen werden, die in einer relevanten Beziehung mit der jeweiligen Rolle stehen. Eine „relevante Beziehung" in diesem Sinne versteht sich als das Ergebnis einer Suche nach der Bezeichnung der Unternehmensrolle (z.B. „Web Developer") in der Menge aller verfügbaren Ausschreibungspositionen. Alle Stellenausschreibungen, die im Suchergebnis enthalten sind, werden als relevant betrachtet und stehen für die weitere Verarbeitung durch ROBUS zur Verfügung. Es muss jedoch darauf geachtet werden, dass Duplikate erkannt und alle Vorkommen insgesamt nur ein Mal gezählt werden. Um ein Duplikat im Kontext dieser Arbeit handelt es sich dann, wenn sich zwei Stellenbeschreibungen nur anhand eines oder mehrerer Metadaten-Felder unterscheiden, die primären Inhaltsfelder jedoch ident sind.
Untersuchungen im Zuge dieser Arbeit haben gezeigt, dass dies vor allem bei Großunternehmen und Firmenketten der Fall ist, die viele gleiche Positionen an unterschiedlichen Orten besetzen wollen. So wurden zum Beispiel im Untersuchungszeitraum 118 Stellenausschreibungen für eine Position als „Shop Assistent" mit identem Titel-, Beschreibungs- und Anforderungstext gefunden. Der einzige Unterschied zwischen all diesen Stellenausschreibungen lag beim Dienstort. Des Weiteren ist es wichtig zu erwähnen, dass die Anzahl *n* der tatsächlich vom System analysierten Ausschreibungstexte durch Parametrisierung einer Obergrenze beschränkt werden kann, wobei die Ergebnisse 1 bis *n* als relevant betrachtet und für die weitere Analyse selektiert und Ergebnisse ab $n + 1$ verworfen werden.
Für die Durchführung der o.a. Selektion nach Rollenbezeichnungen in den Stellenausschreibungen bietet ROBUS ein TF-IDF basiertes Suchverfahren, welches mit Hilfe der Open Source Bibliothek APACHE LUCENE umgesetzt wurde. Sowohl das Verfahren an sich als auch die Bibliothek werden in Abschnitt 0 detailliert beschrieben.
Bei Datenquellen, die eine eigene Implementierung einer Suchfunktion bereitstellen, kann diese das o.a. Verfahren ersetzen. Wie bereits in Abschnitt 5.1.2 erwähnt, verwendet das dieser Arbeit zu Grunde liegende System die Online Plattform LinkedIn als Daten-

quelle für Ausschreibungsdaten. LinkedIn bietet Entwicklern eine eigene Programmierschnittstelle (Job Search API[50]), mittels derer gezielt nach Stellenausschreibungen gesucht werden kann. Im Kontext dieser Arbeit wird die API verwendet, um mit bestimmten Stichwörtern (Rollenbezeichnung) in den Ausschreibungstexten zu suchen und so die relevanten Stellenausschreibungen für jede Rolle zu selektieren. Des Weiteren werden mit Hilfe der API alle benötigten Details (Id, Titel, Beschreibung) der selektierten Ausschreibungen geladen.

5.1.4 Computerlinguistische Analyse von Ausschreibungstexten

Bei Stellenausschreibungen handelt es sich um natürlichsprachliche, unstrukturierte Daten, die im vorgestellten Verfahren mittels computerlinguistischer Methoden analysiert werden. Die Ergebnisse dieser Analysen sind eine unverzichtbare Voraussetzung für alle nachfolgenden Arbeitsschritte (Auswahl und Gewichtung von Rollentermen). Nicht nur ist eine Generierung von Rollenprofilen im hier beschriebenen Sinne ohne vorherige computerlinguistische Verarbeitung unmöglich, auch die Qualität der Profile – und damit die Qualität aller Anwendungen, die diese Rollenprofile einsetzen (kontextsensitives Suchverfahren) - hängt wesentlich von den Analyseergebnissen ab: Werden relevante Termkandidaten in diesem Schritt nicht identifiziert oder fälschlicherweise ausgeschieden, stehen sie auch in allen weiteren Schritten nicht zur Verfügung und fehlen somit bei der Profilbildung.

Bevor allerdings mit der eigentlichen computerlinguistischen Analyse begonnen werden kann, ist es notwendig, die zu untersuchenden Texte zu bereinigen. Die von der LinkedIn API übermittelten Textdaten enthalten zusätzliche Informationen wie Textformatierung,

[50] https://developer.linkedin.com/apis#jobs

Querverweise (Hyperlinks) und andere Steuerzeichen im HTML[51] Format. All diese Zusatzangaben werden von ROBUS durch die Anwendung einer definierten Heuristik entfernt. Der so entstehende Rohtext wird dann an den nächsten Arbeitsschritt weitergeleitet. Im ersten Schritt der computerlinguistischen Verarbeitung wird der bereinigte Rohtext in einzelne Sätze segmentiert. Anschließend wird jeder Satz mit Hilfe eines Tokenizers in separate Tokens geteilt. Sowohl die Satzerkennung als auch das Tokenizing werden in ROBUS unter Zuhilfenahme der OpenNLP[52] Bibliothek implementiert. Beide Komponenten sowie die Verfahren an sich werden in Kapitel 4.4.2 detailliert beschrieben.
Da zur Erstellung von Termvektoren in ROBUS ausschließlich Nomen und Eigennamen relevant sind, wird in einem dritten Schritt jedem Token die entsprechende Wortart zugwiesen. Nur jene Token, die als Nomen oder Eigennamen identifiziert werden, werden als Termkandidaten weitergeleitet, alle anderen werden als nicht relevant betrachtet und an dieser Stelle ausgeschieden. Die Zuordnung der Wortarten wird in ROBUS ebenfalls mit Hilfe der OpenNLP Bibliothek (siehe oben) implementiert. Sie bietet für diese Zwecke einen POS (Part-of-Speech) Tagger, der auf einem Wahrscheinlichkeitsmodell basiert und entsprechend dem Penn Treebank Tag Set[53] alle Nomen mit *NN* (Noun) und alle Eigennamen mit *NP* (Proper Noun) kennzeichnet. Eine detaillierte Beschreibung des Tagging-Verfahrens sowie aller verwendeten Komponenten findet sich in Kapitel 4.4.

---

51 HyperText Markup Language: www.w3c.org

52 Apache OpenNLP Library: http://opennlp.apache.org/

53 http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench/CQP-HTMLDemo/PennTreebankTS.html

5.1.5 Selektion von Termkandidaten

Nachdem alle verfügbaren Termkandidaten mit Hilfe der im vorigen Abschnitt beschriebenen Methoden aus den Ausschreibungstexten extrahiert wurden, bestimmt ROBUS im nächsten Schritt, ob es sich bei einem Term um einen „allgemeinen“ – und damit irrelevanten – oder um einen „spezifischen“ – und damit relevanten – Ausdruck handelt. Diese Bestimmung erfolgt mit Hilfe des DISCO Thesaurus.

DISCO steht für *European Dictionary of Skills and Competences* und ist das Ergebnis eines mehrjährigen internationalen Forschungsprojekts. Der Thesaurus zählt zu den „umfangreichsten Begriffssammlungen für den Bildungs- und Arbeitsmarkt“ und „bietet eine multilinguale und von Expert/innen geprüfte Terminologie für die Klassifizierung, Beschreibung und Übersetzung von Fertigkeiten und Kompetenzen“ (Müller-Riedlhuber & Ziegler 2012a; Müller-Riedlhuber & Ziegler 2012).

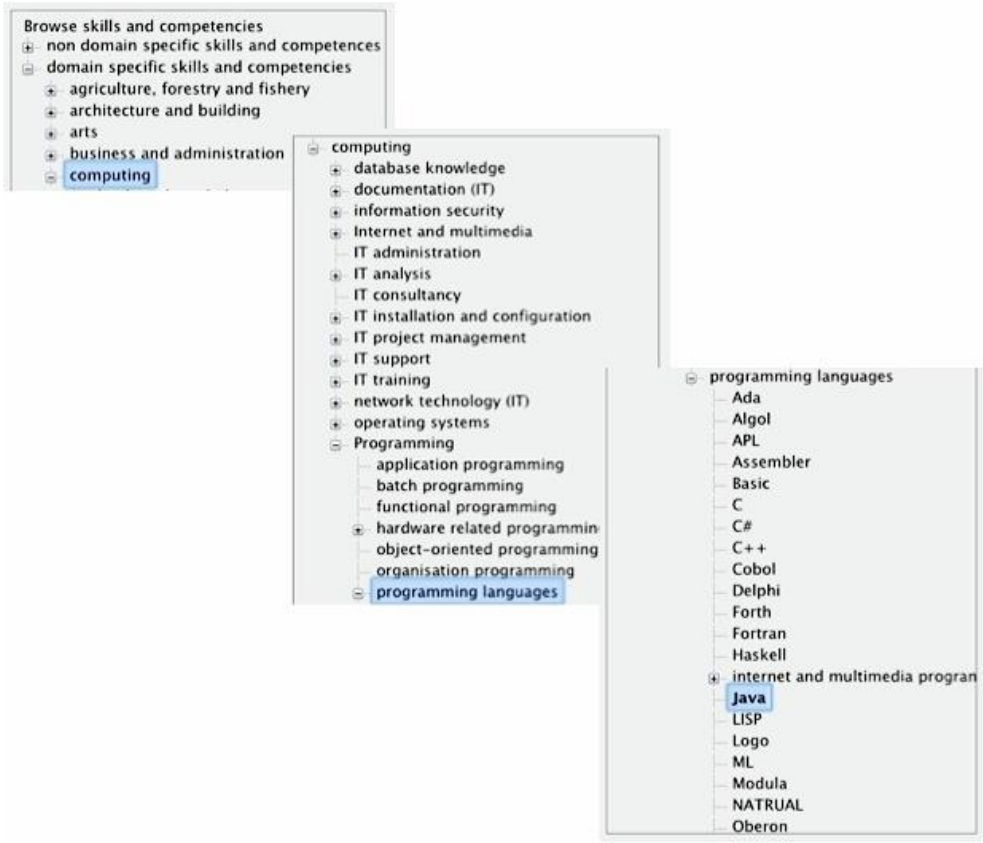

Abbildung 30: Exemplarischer Auszug aus der DISCO Baumansicht anhand des Fertigkeitsbegriffs "Java" aus der Kategorie Domänen-spezifische Fähigkeiten und Kompetenzen → Computing → Programming → Programming Languages → Java (Müller-Riedlhuber & Ziegler 2012b)

Wie oben erwähnt, bestimmt ROBUS mit Hilfe des DISCO Thesaurus, ob es sich bei einem Termkandidaten um einen „allgemeinen" – und damit irrelevanten – oder um einen „spezifischen" – und damit relevanten – Term handelt. Die Unterscheidung erfolgt in der Art, als dass ROBUS den gesamten Bereich der fachlichen (Domänen-spezifischen) Fähigkeiten und Kompetenzen im Thesaurus durchsucht. Ein Termkandidat wird nur dann als relevanter Term betrachtet, wenn dieser auch im Domänen-spezifischen Bereich gefunden wurde. Wird ein Kandidat hingegen nicht oder nur im überfachlichen (Nicht-Domänen-spezifischen) Teil gefunden, interpretiert ROBUS diesen Term als nicht relevant für die weitere Generierung der Termvektoren für die Rollenprofile. Dieses Vorgehen ermöglicht es auf systematische und automatisierte Art und Weise genau jene Begriffe aus Stellenausschreibungstexten zu extrahieren, die eine bestimmte Stelle - und damit Unternehmensrolle - spezifisch beschreiben (z.B. Begriff „Java" für Rolle „Web Developer"), während allgemeine Begriffe, die keine besondere Aussagekraft in Bezug auf eine bestimmte Rolle aufweisen (z.B. Begriff „Ausbildung"), ausgeschieden und bei der weiteren Profilgenerierung nicht mehr betrachtet werden. Die relevanten Terme werden anschließend dem letzten Arbeitsschritt (Gewichtung) zugeführt.

5.1.6 Gewichtung von Rollentermen

Wie bereits erwähnt, bestimmt ROBUS die Relevanz eines natürlichsprachlichen (unstrukturierten) Textes für eine gegebene Unternehmensrolle, indem das System die Ähnlichkeit zwischen dem textuellen Inhalt des Dokuments und dem Rollenprofil ermittelt. Möglich wird hier die Berechnung von Ähnlichkeitswerten durch die Anwendung des VECTOR SPACE MODELS (VSM).

Bei diesem Verfahren werden textuelle Dokumente als gewichtete Termvektoren abgebildet. Jeder Term eines Dokuments entspricht einer Dimension des Vektors mit einem Wert (Gewicht) ungleich null. Für die Berechnung der Gewichte gibt es verschiedenste Methoden. Die wohl bekannteste und am häufigsten verwendete ist die TF-IDF Methode, bei der das Gewicht als Produkt der Termhäufigkeit innerhalb eines Dokuments (TF – Term Frequency) und der umgekehrten Dokumenthäufigkeit (IDF – Inverse Document Frequency) ausgedrückt wird (Manning & Schuetze 1999). Für eine ausführliche Beschreibung sei an dieser Stelle auf Abschnitt 0 verwiesen.

Die Verwendung der TF-IDF Methode setzt jedoch voraus, dass dem System zur Berechnungszeit die gesamte Dokumentensammlung bekannt ist, da ansonsten der IDF Wert nicht korrekt bestimmt werden kann. Dies ist jedoch bei ROBUS nicht der Fall, da Stellenausschreibungen bei Bedarf von Online-Portalen abgerufen werden und somit nicht der gesamte Datenbestand (alle Stellenausschreibungen) sondern nur eine Teilmenge (das Suchergebnis) vorliegt (vgl. Abschnitt 5.1.2). Die Berechnung von Gewichtswerten für Termvektoren ist daher mit der TF-IDF Methode in diesem Fall nicht möglich. Darüber hinaus hat sich gezeigt, dass die Auswertung von Stellenausschreibungen noch weitere Parameter bietet, die bei einer Gewichtsbestimmung berücksichtigt werden können. Daher wird in ROBUS eine neuartige Methode zur Berechnung von Termgewichten eingeführt, die neben der Termhäufigkeit noch die Position des ersten Vorkommens im Text sowie die Hierarchieebene im DISCO Thesaurus berücksichtigt. Außerdem unterscheidet der Algorithmus, in welchen Bereichen der Stellenausschreibung („Title“, „Description“ oder „Skills and Experience“; vgl. Abschnitt 5.1.2) der Term gefunden wurde.

$$w(Zone, Term) = p_D \left(\frac{DiscoLevel_{Term}}{numberLevels_{DISCO}} \right) + p_f \left(\frac{termFreq_{Term}}{numberTerms_{Zone}} \right) + p_p \left(\frac{1 + numberTerms_{Zone} - pos_{Term}}{numberTerms_{Zone}} \right)$$

Gleichung 7: Formel zur Berechnung eines Termgewichts innerhalb eines Bereichs (Zone)

Im ersten Schritt berechnet ROBUS das Gewicht *w* für jeden Term *Term* in einem Bereich *Zone* anhand der Formel in Gleichung 7. Der Parameter $DiscoLevel_{Term}$ beschreibt die Hierarchieebene, auf der der Term im Domänen-spezifischen Bereich des DISCO Thesaurus gefunden wurde. Umso höher der Wert der DISCO Hierarchieebene eines Terms ist, desto spezifischer und damit relevanter ist dieser Term für eine gegebene Rolle. Terme, die nicht im Domänen-spezifischen Bereich des Thesaurus enthalten sind, werden der Ebene 0 zugeordnet. Diese Terme werden von ROBUS aber schon im vorherigen Arbeitsschritt (vgl. Abschnitt 5.1.5) herausgefiltert und spielen daher bei der Berechnung keine Rolle. Zur Normalisierung wird der Wert der Ebene durch die Anzahl aller vorhandenen Ebenen ($numberLevels_{DISCO}$) geteilt. Die Vorkommenshäufigkeit des Terms in der betrachteten Zone wird durch den Parameter $termFreq_{Term}$ ausgedrückt.
Auch dieser Wert wird mittels Division durch die Gesamtanzahl aller Terme in der Zone ($numberTerms_{Zone}$) normalisiert. Die dritte für die Berechnung maßgebliche Größe ist die Position des ersten Vorkommens des Terms in der Zone (pos_{Term}). Umso eher der Begriff genannt wird, desto relevanter wird er von ROBUS für diese Stellenausschreibung erachtet. Dementsprechend führt ein kleinerer Positionswert zu einem höheren Gesamtgewicht und vice versa. Wie bei den beiden Parametern zuvor, erfolgt auch in diesem Falle eine Normalisierung in Bezug auf die Gesamtanzahl der Terme ($numberTerms_{Zone}$).
Ferner enthält die Formel drei Gewichtungsparameter mit deren Hilfe die relative Bedeutung der Thesaurushierarchie (p_D), der Vorkommenshäufigkeit (p_f) sowie der Termposition (p_P) konfiguriert wird.

$$w(Term) = \sum_{i=1}^{N_{Zone}} p_i * w_i, \sum_{i=1}^{N_{Zone}} p_i = 1$$

Gleichung 8: Formel zur Zusammenführung von Termgewichten unterschiedlicher Bereiche (Zonen)

Nachdem die Termgewichte für alle Zonen ermittelt wurden, führt ROBUS in einem zweiten Berechnungsschritt die Gewichte gleicher Terme aus unterschiedlichen Zonen in ein gemeinsames Gewicht *w(Term)* zusammen. In Gleichung 8 findet sich die dafür verwendete Formel, die im Prinzip dem WEIGHTED ZONE SCORE Algorithmus nach (Manning & Schuetze 1999) entspricht: für jede Zone, in der der Term vorhanden ist, wird deren Gewicht mit einem Parameter *p* multipliziert und zum Gesamtgewicht addiert. Mithilfe des Parameters *p* können Zonen als unterschiedlich wichtig definiert werden. So ist es zum Beispiel möglich, der Titelzone einen höheren Parameterwert und damit eine höhere Relevanz zuzuschreiben, als der Beschreibungszone. Es ist jedoch darauf zu achten, dass die Summe aller Parameterwerte stets 1 ergibt.

Im dritten Schritt des Gewichtungsverfahrens werden gleiche Terme aus unterschiedlichen Ausschreibungstexten gruppiert. Dies geschieht durch Zusammenlegen gleicher Termeinträge und Addition der entsprechenden Gewichtswerte.

Abschließend generiert ROBUS die zur Repräsentation der Rollenprofile verwendeten Termvektoren, indem es jeden ermittelten Term und dessen zugehöriges Gewicht als Term-Gewicht-Tupel in den Vektor einfügt. Im Zuge der Systemkonfiguration wird die maximale Länge n_{Max} der Profilvektoren bestimmt. Standardmäßig enthält jeder Vektor zwanzig Term-Gewicht-Tupel ($n_{Max} = 20$). Sollte für ein Rollenprofil die Anzahl der identifizierten Termkandidaten n_{Max} übersteigen, so werden nur die n Terme mit den höchsten Gewichtswerten hinzugefügt. Alle weiteren Einträge werden verworfen und für die Profilgenerierung nicht weiter berücksichtigt.

Nachfolgende Tabelle zeigt exemplarisch mehrere, von ROBUS generierte Terme und deren zugehörige Gewichtswerte für die Unternehmensrollen (1) „Web Developer“, (2) „Marketing Director“ und (3) „Network Engineer“.

(1) Web Developer		**(2) Marketing Director**		**(3) Network Engineer**	
Term	*Gewicht*	*Term*	*Gewicht*	*Term*	*Gewicht*
web	3.05748	marketing	2.46174	network	2.31579
developer	2.56737	company	1.57357	lan	2.09037
asp	1.83625	outbound	1.55121	engineer	2.07981
php	1.74986	xml	1.50824	design	1.7415
javascript	1.56781	search	1.43294	ethernet	1.31905
application	1.51373	sales	1.42044	wan	1.17076
c#	1.48317	engine	1.41613	engineering	1.15824
http	1.46808	market	1.36696	access	1.14854
information	1.35218	media	1.35231	systems	1.11115
sql	1.34565	mobile	1.3473	enterprise	1.07941
work	1.31551	team	1.25207	technology	1.0462
race	1.26924	management	1.20987	security	1.04148
design	1.22719	leadership	1.19834	isdn	1.02572

Tabelle 11: von ROBUS generierte Termvektoren für die Rollen "Web Developer", "Marketing Director" und "Network Engineer"

Die auf diese Art und Weise generierten Profilvektoren können für verschiedenste Kontext-bezogene Anwendungsfälle verwendet werden. Neben dem nachfolgend im Detail geschilderten Einsatzbereich der Rollen-sensitiven Suche, können Termvektor-basierte Rollenprofile in Zukunft vor allem im Umfeld von Enterprise 2.0 (zum Beispiel zur Optimierung und Evaluierung von sozialen Tags und Bookmarks) eine bedeutende Stellung einnehmen. Die gegenständliche Arbeit legt jedoch einen klaren Fokus auf den Einsatz von Rollenprofilen zur kontext-sensitiven Suche und beschäftigt sich nicht eingehender mit anderen Anwendungsfällen.

5.2 Rollen-sensitive Suche

Wie eingangs erwähnt, verfolgt ROBUS den Zweck, die unternehmensweite Suche in natürlichsprachlichen (unstrukturierten) Datenbeständen zu verbessern, indem der Kontext und die langfristigen Informationsbedürfnisse der Benutzer/innen in Form ihrer Unternehmensrolle berücksichtigt werden. Dies geschieht in ROBUS durch den Einsatz von Rollenprofilen (siehe voriger Abschnitt), die jedem/r Benutzer/in zugeordnet sind. Die Rollen-sensitive Suche in ROBUS verwendet die gewichteten Termvektoren der Profile und ermittelt anhand derer die Relevanz jedes einzelnen Dokuments für jede im Unternehmen definierte Rolle. Auf Basis dieser Relevanzwerte sortiert ROBUS die Reihung der Suchergebnisse, sodass stets jene Dokumente an vorderster Stelle erscheinen, die die höchste Relevanz für die jeweiligen Benutzer/innen aufweisen. Sowohl das Verfahren zur Relevanzbestimmung als auch die Methode der Rollen-sensitiven Reihung werden in den beiden nachfolgenden Abschnitten detailliert beschrieben.

5.2.1 Berechnung der Rollenrelevanz

Zur Berechnung der Rollenrelevanzwerte folgen wir dem Verfahren wie in (Reichhold et al. 2011) beschrieben: Um eine Rollen-sensitive Suche durchgehend anwenden zu können, ist es erforderlich, für jedes bekannte Dokument in der Datensammlung des Unternehmens einen Rollenrelevanzvektor

$$RR_d = (RS_{r1}, RS_{r2}, \dots , RS_{rn})$$

zu erstellen. Jeder Rollenrelevanzvektor enthält wiederum genau einen Relevanzwert *RS* für jede im Unternehmen definierte Rolle *r*. Der Relevanzwert *RS* gibt an, wie relevant ein Dokument *d* für die Rolle *r* ist und ist somit die grundlegende Basis für die Rollen-sensitive Reihung von Suchergebnissen in ROBUS. Ermittelt wird der Relevanzwert

durch Bestimmung der Ähnlichkeit zwischen dem jeweiligen Dokument und dem gewichteten Termvektor des Rollenprofils, wobei gilt, dass ein Dokument umso relevanter für eine Rolle ist, desto höher seine Ähnlichkeit zum Termvektor ist.
Zur Berechnung der Ähnlichkeit sind in der Literatur verschiedenste Maße und Verfahren bekannt (für eine ausführliche Beschreibung siehe Abschnitt 2.3). In ROBUS wird an dieser Stelle das Vector Space Model und die Kosinus-Ähnlichkeit angewandt. Dabei wird nicht nur das Rollenprofil, sondern auch das Dokument selbst als gewichteter Termvektor interpretiert, wobei jeder vorkommende Term eine Dimension des Vektors repräsentiert. Die Ähnlichkeit hierbei ist definiert als Kosinus des Winkels zwischen den beiden Vektoren und wird mittels nachstehender Formel berechnet.

$$RS_r = \cos\theta = \frac{T_d * RT_r}{|T_d| * |RT_r|}$$

Gleichung 9: Berechnung der Kosinus-Ähnlichkeit zwischen dem Vektor T_d eines Dokuments *d* und dem *Termvektor* RT_r der Rolle *r*

Dabei entspricht T_d der Vektorrepräsentation des Dokuments *d* und RT_r dem Termvektor der Rolle *r*. Umso kleiner der Winkel zwischen den beiden Vektoren, desto ähnlicher sind sie einander und desto höher ist die Relevanz des Dokuments für die Rolle in ROBUS. Ein Ergebnis von 1 entspricht einem Winkel von 0 und bedeutet somit, dass die beiden Vektoren genau übereinander liegen. Ein Ergebnis von -1 würde bedeuten, dass die Vektoren in die exakt gegenüberliegende Richtung zeigen. Da in ROBUS allerdings keine negativen Gewichtswerte vorkommen können, ist der kleinstmögliche Wert 0. Die Länge des Vektors hat hingegen keine Auswirkung auf die Bestimmung der Ähnlichkeit. Das ist ein großer Vorteil bei der Anwendung der Kosinus-Ähnlichkeit in ROBUS, da somit auch die Größe (bzw. Länge) der Textdokumente nicht weiter ausschlaggebend ist.

5.2.2 Reihung von Suchergebnissen

Heutige Unternehmenssuchmaschinen berücksichtigen zumeist keinerlei kontextuelle Informationen. Sie verarbeiten ausschließlich die von den Mitarbeiter/innen in Form von Schlüsselwörtern eingegebenen Suchanfragen und liefern daraufhin ein einheitliches Suchergebnis. Auf die unterschiedlichen Informationsbedürfnisse der einzelnen Benutzer/innen wird dabei nicht Rücksicht genommen; jede/r Benutzer/in erhält für die gleiche Anfrage genau das gleiche Suchergebnis. Dieser Einschränkung wird mit der Rollen-sensitiven Suche in ROBUS entgegengewirkt, indem das Suchergebnis der zugrunde liegenden nicht Rollen-sensitiven Suche neu gereiht wird. Diese Umreihung erfolgt in Abhängigkeit der Relevanz der Dokumente in Bezug auf die Rolle, die dem suchenden Benutzer zugeordnet ist, wobei Dokumente mit höheren Relevanzwerten weiter nach oben gereiht werden und umgekehrt. Durch diese Vorgehensweise wird gewährleistet, dass die langfristigen Informationsbedürfnisse der einzelnen Mitarbeiter/innen bei der Suche einfließen und das „One-Fits-4-All" Problem behoben wird.

Die Umsetzung dieser Vorgehensweise passiert mittels eines speziellen Algorithmus, der den Rollen-sensitiven Ergebniswert mit dem originalen (nicht Rollen-sensitiven) Wert zusammen führt. Vorlage für diesen Algorithmus war die Arbeit von (E Agichtein et al. 2006), die viele verschiedene Ansätze evaluiert haben und schließlich fanden, dass „eine einfache Heuristik zur Zusammenführung von Rängen sehr gute Ergebnisse liefert und darüber hinaus eine hohe Robustheit gegenüber Variationen von Bewertungswerten unterschiedlicher Suchmaschinen aufweist".

Die in (E Agichtein et al. 2006) vorgestellte Gleichung wurde für den Einsatz in ROBUS, wie in Gleichung 10 dargestellt, adaptiert. Der neue, Rollen-sensitive Rang ergibt sich aus dem Rang der Rollenrelevanzvektoren R_d sowie dem Rang der originalen, nicht kontext-sensitiven Suche O_d. Der Rollenrelevanz-Rang R_d wiederum ergibt sich aus der Reihung (in absteigender Sortierung) der Dokumente anhand ihres Relevanzwertes für das jeweilige Rollenprofil.

$$S(R_d, O_d, w) = w * \frac{1}{R_d + 1} + (1 - w) * \frac{1}{O_d + 1}$$

Gleichung 10: Ermittlung des zusammengeführten Rangs anhand des originalen und des Rollen-sensitiven Suchergebnisses

Der Parameter *w* dient zur Einstellung des Gewichtsverhältnisses zwischen dem originalen und dem Rollen-sensitiven Rang. Ein Wert von 0,5 bedeutet eine gleichwertige Verteilung der beiden Ränge.

5.3 Zusammenfassung

In diesem Kapitel wurde gezeigt, wie ROBUS unternehmensweite Rollenprofile erzeugt und diese zur Optimierung der Suche in natürlichsprachlichen Daten einsetzt. Abbildung 31 zeigt einen schematischen Gesamtüberblick der Rollen-sensitiven Suche in ROBUS inklusive der Generierung der für die Suche benötigten Termvektoren.
Im Abschnitt 5.1 wird beschrieben, wie das System völlig automatisiert Rollenprofile für alle in einem Unternehmen definierten Mitarbeiterrollen erzeugt und wie diese Rollenprofile als gewichtete Termvektoren repräsentiert werden. Ferner wird dargestellt, wie anhand von Rolleninformationen, die Mitarbeiter/inne/n zugeordnet sind, ein Zusammenhang zwischen den langfristigen Informationsbedürfnissen der Benutzer/innen und den textuellen, unstrukturierten Inhalten der Unternehmensdaten hergestellt werden kann.
Als grundlegender Ausgangspunkt dafür werden Stellenausschreibungen präsentiert.

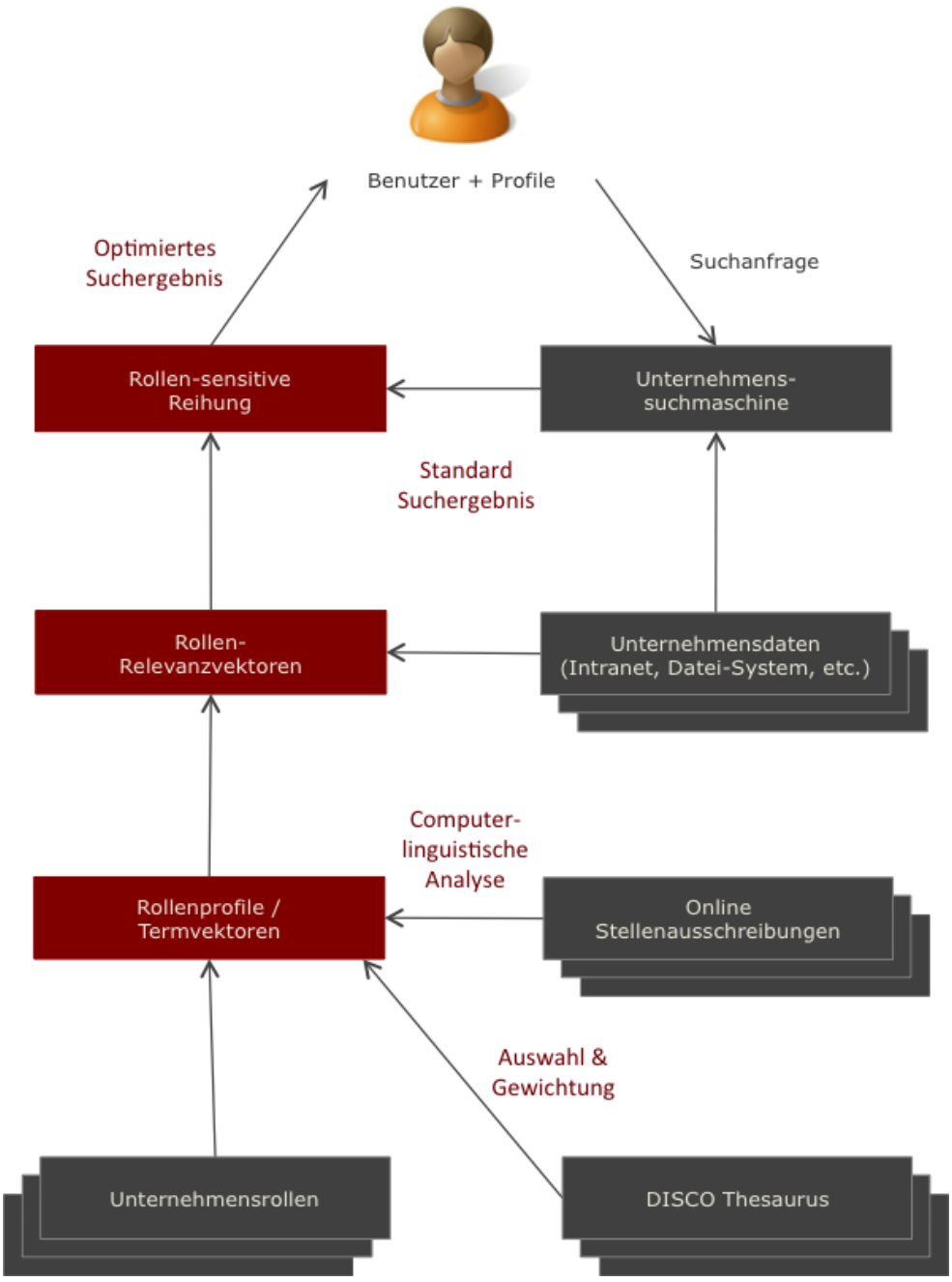

Abbildung 31: Schematischer Gesamtüberblick der Rollen-sensitiven Suche in ROBUS

Die Analyse und Verarbeitung der in natürlicher Sprache formulierten Beschreibungstexte mit Hilfe von computerlinguistischer Methoden ist eine der wesentlichen Komponenten des ROBUS Systems. In Abschnitt 5.1.4 ist im Detail festgehalten, wie ROBUS nach einer initialen Bereinigung (Entfernen von Formatierungs- und Steuerzeichen) den

eingehenden Textfluss in einzelne Sätze und Tokens aufteilt, und wie das System anschließend für jeden Token eine Wortart bestimmt. Auf Basis dieser Analyseergebnisse werden mit Hilfe des DISCO Thesaurus Termkandidaten ausgewählt und an die Gewichtungsfunktion weitergeleitet (siehe Abschnitt 5.1.5).
Der finale Schritt im Zuge der Rollenprofilerstellung ist die Gewichtung der vorab ausgewählten Termkandidaten. Die konkrete Vorgehensweise zur Erstellung der Termvektoren sowie die eigens für ROBUS entwickelte Gewichtungsfunktion werden in 5.1.6 spezifiziert.
Abschnitt 5.2 beschreibt schließlich, wie Rollenprofile in ROBUS eingesetzt werden, um die Relevanz eines Dokuments für eine bestimmte Unternehmensrolle mit Hilfe des Vektorraum-Models und der Kosinus-Ähnlichkeit zu berechnen. Der für jedes Dokument und jede Rolle im Unternehmen definierte Relevanzwert wird im Zuge einer Suchanfrage von einem speziellen Sortieralgorithmus angewandt, um die vorhandenen Suchergebnisse in Bezug auf die Relevanz für die Rolle des suchenden Mitarbeiters neu zu reihen. Durch dieses Vorgehen wird letztendlich das eigentliche Ziel von ROBUS - eine Rollen-sensitive Suche für unstrukturierte Unternehmensdaten – umgesetzt.

6 Evaluationsmethode & Ergebnisse

Im nachfolgenden Kapitel wird die konkrete Methode - inklusive aller dabei verwendeten Ressourcen und Komponenten - beschrieben, die zur Evaluation von ROBUS entwickelt und eingesetzt worden ist. So werden neben den eigentlichen Testdaten (Testkorpus, personalisierte Suchanfragen und Relevanzbeurteilungen) auch die spezifischen Evaluationsmetriken und die benutzten Vergleichssysteme (Baseline-Systeme) detailliert beschrieben. Des Weiteren werden die für die Evaluation von ROBUS generierten Rollenprofile sowie alle relevanten Testparameter im Detail dokumentiert, sodass sämtliche Ergebnisse und Rückschlüsse des Autors nachvollzogen werden können.
Im zweiten Teil des Kapitels werden die finalen Evaluationsergebnisse des ROBUS Systems präsentiert. Nach einer einführenden Erläuterung der zugrundeliegenden Testkonfiguration werden die konkrete Testdurchführung und anschließend die daraus resultierenden Ergebnisse beschrieben. Der dritte und abschließenden Teil des Kapitels enthält eine Zusammenfassung und Diskussion der Ergebnisse.

6.1 Evaluationsmethode für ROBUS

Wie in Kapitel 0 ausführlich erläutert, ist ROBUS ein Informationssuchsystem, das darauf abzielt, die Suche innerhalb unstrukturierter, natürlichsprachlicher Unternehmensdaten zu verbessern, indem kontextuelle Informationen über den suchenden Benutzer berücksichtigt werden. Da dementsprechend unterschiedliche Benutzer/innen unterschiedliche Suchergebnisse trotz Eingabe einer identen Suchanfrage (engl. Query) erhalten, kann ROBUS auch aus Sicht der Evaluation als personalisiertes Suchsystem betrachtet werden. Die Möglichkeiten und Verfahren zur Evaluation von Suchsystemen im Allgemeinen und zur Evaluation von personalisierten Systemen im Speziellen wurden im vorhergehenden Kapitel (0) detailliert beschrieben.

Bevor die eigentlichen Evaluationsergebnisse für ROBUS präsentiert werden, soll an dieser Stelle noch die konkrete Vorgehensweise sowie die dabei verwendeten Komponenten bei der Durchführung beleuchtet werden. Für die Evaluation von ROBUS wurde eine standardisierte Sammlung an Dokumenten herangezogen, die mit Hilfe des Folksonomy Ansatzes um personalisierte Suchanfragen und Relevanzbeurteilungen erweitert wurde. Diese Vorgehensweise hat es ermöglicht, ein umfangreiches Korpus für die Evaluation von personalisierten Suchsystemen zu generieren, ohne dabei auf die aufwändige manuelle Erstellung von Relevanzbeurteilungen durch Domänenexperten beziehungsweise Testpersonen angewiesen zu sein. Auch bei der Auswahl der Bewertungsmetrik wurde darauf geachtet, nicht nur ein möglichst passendes, sondern auch standardisiertes und weit verbreitetes Kennzahlensystem zu verwenden, sodass die Vergleichbarkeit und Transparenz gegenüber anderen Systemen und Testergebnissen gewahrt bliebe.

Im Zuge der Evaluationsdurchführung des ROBUS Systems wurden die Testergebnisse einem äußerst kompetitiven Vergleichssystem (engl. Baseline) gegenübergestellt, wodurch gezeigt werden konnte, dass ROBUS trotz seiner noch frühen Entwicklungsphase bereits wichtige Beiträge zur Verbesserung der Sucheffektivität liefert. Ausgehend von diesen Informationen werden im Folgenden alle Methoden und Komponenten, die zur Evaluation von ROBUS eingesetzt wurden, im Detail vorgestellt.

6.1.1 Anforderungen an das Testkorpus

Zur Evaluation von Informationssuchsystemen kann entweder eine eigene Datensammlung generiert oder ein bestehender, standardisierter Testkorpus verwendet werden. Seit Entwicklung der ersten Datensammlungen im Zuge der Cranfield Experimente in den 1960er Jahren wurden eine Vielzahl von standardisierten Testkorpora zu Evaluationszwecken geschaffen. Zu den bekanntesten und am weitesten verbreiteten gehören wohl jene der TREC Reihe (vgl. Kapitel 3.3). Im Allgemeinen bietet die Verwendung eines standardisierten Korpus mehrere Vorteile gegenüber der Erstellung eines eigenen Korpus, wie beispielsweise ein deutlich geringerer Aufwand bei der Erstellung der Testdaten, bessere Nachvollziehbarkeit sowie höhere Vergleichbarkeit mit anderen Systemen und Testergebnissen, weshalb die Präferenz desselben begründet erscheint.
Die wichtigsten Kriterien bei der Auswahl eines Korpus zur Evaluation von Informationssuchsystemen sind neben der Größe der Datensammlung (d.h. die Anzahl der enthaltenen Elemente) und dem Inhalt der Dokumente, die definierten Informationsbedürfnisse beziehungsweise Suchanfragen und die zugehörigen Relevanzbeurteilungen. Es ist bei der Selektion des Korpus dementsprechend darauf zu achten, dass sowohl die Dokumentinformationen als auch die Suchanfragen und Relevanzbewertungen den spezifischen Anforderungen des zu evaluierenden Systems genügen, und dass sämtliche Elemente in ausreichender Zahl vorhanden sind. Unter spezifischen Anforderungen ist in diesem Sinne vor allem gemeint, unter welchen Umständen und Rahmenbedingungen beziehungsweise in welchen Anwendungsgebieten das Suchsystem eingesetzt werden soll. So wurden beispielsweise im Zuge der TREC Reihe spezielle Testdatensätze zur Evaluation von Frage-Antwort-Systemen („Question Answering Track"), Suchsystemen für neue Internetinhalte („Blog Track") und Suchaufgaben innerhalb von Organisationen („Enterprise Track") veröffentlicht (Sanderson 2010). Eine weitreichende Übersicht verschiedener standardisierter Testkorpora sowie deren Eigenschaften und Einsatzmöglichkeiten gibt (Sanderson 2010); weitere Details siehe Kapitel 0.
Sanderson beschreibt aber auch die Limitierungen von standardisierten Testdatensätzen und führt als Beispiel Testergebnisse aus dem TREC-9 Web Track an, wonach zum Bei-

spiel der etablierte Internet PageRank[54] Algorithmus zu keiner signifikanten Verbesserung der Sucheffektivität beigetragen hätte. Untersuchungen ergaben, dass der Grund dafür darin lag, dass Suchanfragen im Internet häufig als Navigationsanfragen[55] vorkommen und sich dadurch sehr stark von den „klassischen" Suchanfragen, die aus den TREC Informationsbedürfnissen („Topics") abgeleitet wurden, unterscheiden. Navigationssuchanfragen waren aber im TREC Korpus nicht existent und wurden somit bei der Evaluation nicht berücksichtigt. Dieses Beispiel zeigt sehr deutlich, welchen starken Einfluss die Auswahl des Testkorpus beziehungsweise das Fehlen wichtiger Elemente auf die Evaluationsergebnisse hat.

Die Evaluation von kontext-sensitiven und personalisierten Suchsystemen, d.h. Systeme, die neben der eingegebenen Suchanfrage noch weitere Faktoren wie beispielsweise Benutzerprofile berücksichtigen, stellt eine besondere Herausforderung dar. Diese Systeme betrachten die Relevanz eines Dokuments in Bezug auf eine Suchanfrage nicht als objektive Eigenschaft des Dokuments. Vielmehr berücksichtigen sie die definierten kontextuellen Parameter und liefern abhängig davon unterschiedliche Suchergebnisse für ein- und dieselbe Suchanfrage. Folglich müssen auch die Relevanzbeurteilungen der verwendeten Evaluationskorpora diesem Umstand Rechnung tragen und die kontextuellen Parameter berücksichtigen.

Wie bereits in Kapitel 3.5 erläutert, handelt es sich bei der Generierung von kontextsensitiven beziehungsweise personalisierten Suchanfragen und Relevanzbeurteilungen um einen nicht-trivialen Vorgang. Die Bereitstellung von wiederverwendbaren standardisierten Datensammlungen mit solchen Informationen ist ungleich schwieriger und die Verfügbarkeit dementsprechend eingeschränkt oder gar nicht gegeben. In Kapitel 3.5

---

[54] PageRank ist ein populärer Algorithmus der ursprünglich von Sergey Brin and Lawrence Page an der Universität Stanford für ihre Internetsuchmaschine Google entwickelt wurde. Details siehe: http://infolab.stanford.edu/~backrub/google.html

[55] Unter Navigationssuchanfragen (engl. Navigational Queries) versteht man Anfragen in Internetsuchmaschinen, bei denen der Benutzer eine bestimmte Seite (Homepage) sucht.

werden zwar mehrere Vorgehensweisen und Datensammlungen zur Evaluation von personalisierten Suchsystemen angeführt. Es existiert jedoch keine Datensammlung, die alle Anforderungen zur Evaluation von ROBUS abdeckt und auch verfügbar ist.
Das Modell von (Xu et al. 2008) beschreibt die Generierung von personalisierten Suchanfragen und Relevanzbeurteilungen, aber weder das in der Arbeit beschriebene resultierende Testkorpus noch die zugrundeliegenden Datensätze („IBM Dogear“) sind verfügbar[56]. Ähnlich verhält es sich mit der Arbeit von (Harpale et al. 2010). Das darin beschriebene CiteData Korpus wurde explizit für die Evaluation von personalisierten Suchsystemen entwickelt (vgl. Kapitel 3.7). Ein Großteil der verwendeten Komponenten, darunter Dokumentinformationen, Suchanfragen und personalisierte Relevanzbewertungen, stehen auch zum Download zur Verfügung. Es fehlen jedoch essentielle Elemente (Informationen zum Ableiten von Benutzerprofilen), die entgegen der Ankündigung der Autoren, bisher auch noch nicht veröffentlicht wurden. Eine persönliche Nachfrage bei den Autoren blieb leider unbeantwortet.
Da kein geeignetes standardisiertes Testkorpus zur Evaluation von ROBUS verfügbar war, wurde auf Grundlage der in (Harpale et al. 2010) und (Xu et al. 2008) beschriebenen Methoden eine eigene Datensammlung für diesen Zweck erstellt. Die dabei angewandte Vorgehensweise und deren Ergebnisse werden in den folgenden Kapiteln detailliert beschrieben.

6.1.2 Dokumente der Testsammlung

Die Grundlage jedes Textkorpus ist eine Sammlung von Datensätzen, die in aller Regel in Form von einzelnen Dokumenten vorliegen. Der Inhalt dieser Dokumente sollte die Gegebenheiten des Anwendungsgebietes, für das das evaluierte System entwickelt wird,

[56] Auf persönliche Anfrage des Autors zur Bereitstellung der beschriebenen Daten bei den verantwortlichen Entwicklern von (Xu et al. 2008) und „IBM Dogear“ wurde hingewiesen, dass diese aus (datenschutz-)rechtlichen Gründen nicht veröffentlicht werden können

möglichst gut widerspiegeln. Auch die Anzahl der vorhandenen Dokumente sollte ein bestimmtes Mindestmaß aufweisen, um ein repräsentatives Abbild darzustellen und damit eine höhere Aussagekraft der Evaluation zu erreichen. Eine konkrete Mindestmenge an Dokumenten wird in der Literatur nicht genannt. Manning et al. argumentieren jedoch in ihrer Arbeit, dass beispielsweise die 1398 Dokumente, die der ursprüngliche Cranflield Testkorpus (vgl. Kapitel 3.2) enthielt, heutzutage nur noch für die einfachsten und grundlegendsten Experimente herangezogen werden kann. Für eine umfangreiche und aussagekräftige Evaluation ist diese Dokumentensammlung aber viel zu klein (Manning et al. 2008).

Für die ROBUS Testdatensammlung stellte „CiteULike“ die Ausgangsbasis dar. Diese Website ist unter http://www.citeulike.org/ im Internet erreichbar und frei verfügbar. Die Benutzer/innen dieser Website können sich akademische Publikationen, die für sie von besonderem Interesse sind, merken (engl. Bookmark) und darüber hinaus für jeden gemerkten Artikel beliebig viele frei definierbare Schlagwörter (engl. Tags, Social Tags oder auch Social Annotations) vergeben. Sie enthält eine Sammlung von aktuell über vier Millionen akademischen Publikationen aus den unterschiedlichsten Fachbereichen und kann über 124.000 registrierte und aktive Benutzer/innen aufweisen. Als aktive Benutzer/innen in diesem Sinne gelten alle Benutzer/innen, die zumindest eine Publikation in CiteULike verlinkt beziehungsweise zumindest ein Schlagwort (Tag) vergeben haben.

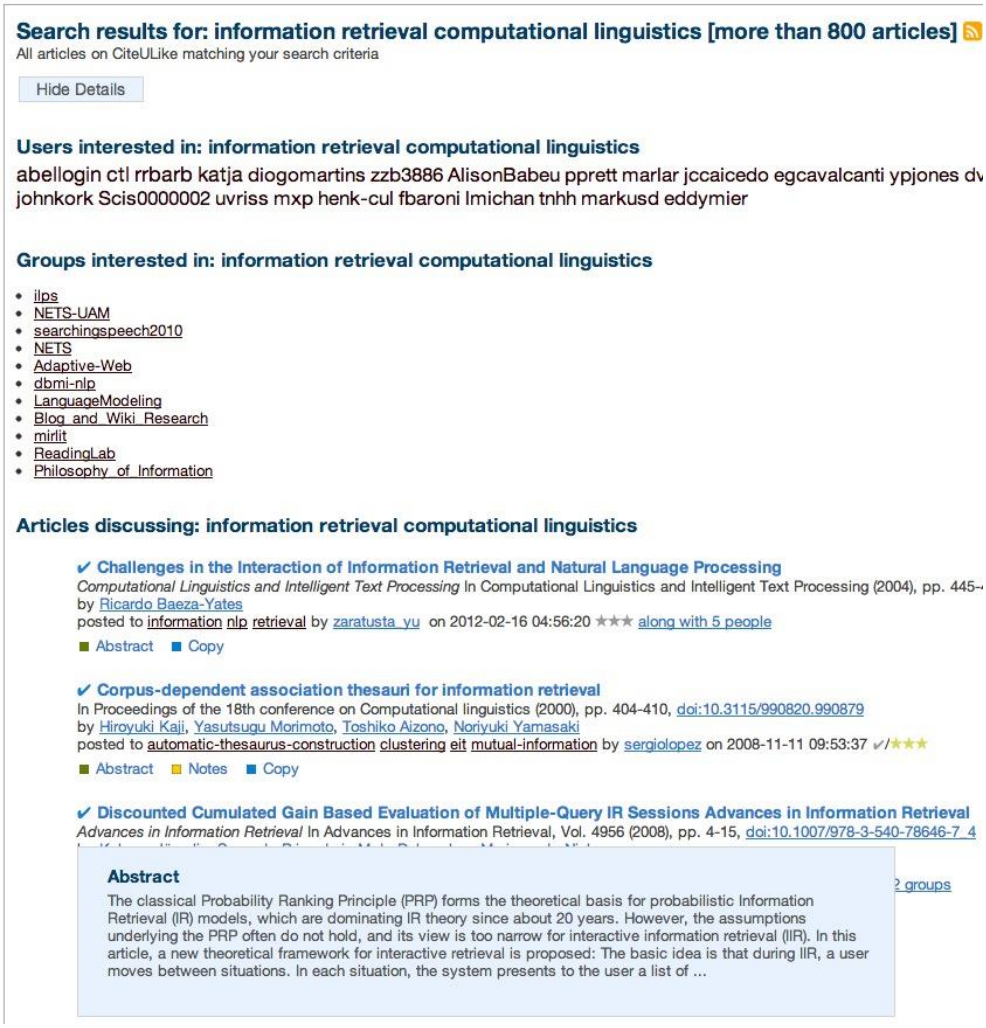

Abbildung 32: Auszug des Suchergebnisses für die Suche nach INFORMATION und RETRIEVAL und COMPUTATIONAL und LINGUISTICS in CiteULike (durchgeführt am 27.07.2013, auf www.citeulike.org)

Der Großteil der Publikationen ist in englischer Sprache verfasst. Das System selbst erlaubt aber der Bereitstellung von Artikeln in verschiedensten Sprachen. Dementsprechend existieren auch zahlreiche Publikationen in anderen Sprachen wie beispielsweise Deutsch, Französisch oder Spanisch. Jede Publikation im CiteULike System enthält neben dem Titel noch eine Reihe von zusätzlichen Informationen. Zu diesen Metadaten zählen (1) der/die Autor/in beziehungsweise die Autor/inn/en, (2) die Liste der Editor/inn/en, (3) das Veröffentlichungsmedium sowie (4) weitere Dokumentdetails zur

Identifikation des Elements (DOI[57] ID, CiteULike Key) angezeigt. Viele Artikel verfügen auch über eine Kurzfassung (Abstract) des Dokuments. Diese Information ist jedoch kein Pflichtfeld und daher nicht bei allen Artikeln vorhanden. Abbildung 32 zeigt, dass in CiteULike mehr als 800 Artikel zur Suchanfrage „INFORMATION und RETRIEVAL und COMPUTATIONAL und LINGUISTICS“ gefunden werden. Des Weiteren werden die ersten drei Suchergebnisse mit ihren Metainformationen (Autoren, Editoren, Veröffentlichungsmedium, Veröffentlichungsdatum und zugewiesene Schlagworte) dargestellt.
Aus dem Datenpool der vier Millionen CiteULike Artikel wurden insgesamt 77.000 Publikationen für das ROBUS Testkorpus ausgewählt. Die Auswahl der Artikel erfolgte nach dem Zufallsprinzip. Es wurden aber ausschließlich Artikel mit englischen Textinhalten berücksichtigt und es wurden nur Artikel in das Korpus aufgenommen, die über eine aussagekräftige Kurzbeschreibung (Abstract) verfügen. Der Export der Artikeldaten aus dem CiteULike System erfolgte mithilfe der vom System angebotenen Web-Schnittstelle im JSON[58] Format (CiteULike 2012).
Abbildung 33 zeigt einen Ausschnitt eines für das Korpus verwendeten Artikels mit seinen Metainformationen „id“ (CiteULike Kennung), „content_abstract“ (Kurzbeschreibung), „path“ (Pfad zum Artikel auf der CiteULike Website), „title“ (Titel) und „doi“ (DOI Kennung). Zur weiteren Verwendung der 77.000 Datensätze im Korpus wurden alle Dokumente in einer relationalen SQL[59] Datenbank gespeichert. Nach dem eigentlichen Export der Artikelinformationen wurde noch eine Datenbereinigung mithilfe eines selbsterstellten Regelwerks durchgeführt. Dabei wurden alle Duplikate (Artikel mit gleichem Titel) und Dokumente ohne aussagekräftigen Abstract (Abstract ist kürzer als 200 Zeichen, oder Abstract ident mit Titel) ausgeschlossen.

---

[57] DOI ... Digital Object Identifier: http://www.doi.org/

[58] JSON ... JavaScript Object Notation: http://json.org/

[59] SQL ... Structured Query Language: http://www.iso.org/iso/iso_catalogue/catalogue_tc/ catalogue_tc_browse.htm?commid=45342

```
{⊟
   "data": [⊟
      {⊟
         "id": "6194667",
         "content_abstract": "The Pandemic (H1N1) 2009 is spreading to numerous countries and c
ausing many human deaths. Although the symptoms in humans are mild at present, fears are that f
urther mutations in the virus could lead to a potentially more dangerous outbreak in subsequent
 months. As the primary immunity-eliciting antigen, hemagglutinin (HA) is the major agent for h
ost-driven antigenic drift in A(H3N2) virus. However, whether and how the evolution of HA is in
fluenced by existing immunity is poorly understood for A(H1N1). Here, by analyzing hundreds of
A(H1N1) HA sequences since 1918, we show the first evidence that host selections are indeed pre
sent in A(H1N1) HAs. Among a subgroup of human A(H1N1) HAs between 1918~2008, we found strong d
iversifying (positive) selection at HA1 156 and 190. We also analyzed the evolutionary trends a
t HA1 190 and 225 that are critical determinants for receptor-binding specificity of A(H1N1) HA
. Different A(H1N1) viruses appeared to favor one of these two sites in host-driven antigenic d
rift: epidemic A(H1N1) HAs favor HA1 190 while the 1918 pandemic and swine HAs favor HA1 225. T
hus, our results highlight the urgency to understand the interplay between antigenic drift and
receptor binding in HA evolution, and provide molecular signatures for monitoring future antige
nically drifted 2009 pandemic and seasonal A(H1N1) influenza viruses.",
         "path": "http://www.citeulike.org/user/zwang/article/6194667",
         "title": "Evolutionary Trends of A(H1N1) Influenza Virus Hemagglutinin Since 1918",
         "doi": "10.1371/journal.pone.0007789",
```


Abbildung 33: Exemplarische Abbildung eines aus CiteULike exportierten Artikeldatensatzes (Auszug) im JSON Format

6.1.3 Suchanfragen und Relevanzbeurteilungen

Wie in den vorherigen Kapiteln erwähnt, gibt es verschiedene Methoden und Technologien zum Generieren von Suchanfragen und Relevanzbeurteilungen für die Evaluation von Informationssuchsystemen. Wie ebenfalls bereits erläutert, existiert eine Vielzahl von verfügbaren standardisierten Testkorpora, die nach dem Vorbild der Cranfield Experimente erstellt wurden (vgl. Kapitel 3.2 und 3.3). Diese können aber für die Evaluation von ROBUS nicht herangezogen werden, da sie keine personalisierten Relevanzbeurteilungen und Informationen über die Beurteiler selbst enthalten. Der Einsatz von Testframeworks auf Basis von Query Log Daten (vgl. Kapitel 3.4) scheitert wiederum an der nicht gegebenen Verfügbarkeit der Daten für die gegenständliche Arbeit.
Aus diesem Grund wurde für die Evaluation von ROBUS ein Folksonomy-basierter Ansatz gewählt. Die Auswertung solcher Daten zur Generierung von personalisierten Suchanfragen und Relevanzbeurteilungen hat in den letzten Jahren stark an Bedeutung gewonnen. Zwei konkrete Vorgehensweisen dazu sind in den Kapiteln 3.6 und 3.7 detailliert beschrieben. Wie schon bei den zuvor erwähnten Dokumenten, stellte auch bei den Suchanfragen und Relevanzbeurteilungen das CiteULike Service die Datengrundlage dar. In

diesem Falle mussten die Daten aber nicht über eine Schnittstelle exportiert werden. Vielmehr stellt CiteULike diverse Datensammlungen zur Verfügung. Diese sind für akademische Zwecke frei erhältlich und können auf der CiteULike Datasets Website[60] nach kostenloser Registration heruntergeladen werden.

Für das ROBUS Testkorpus wurde die „Who-posted-what“ Datensammlung (Stand: 02.01.2013) verwendet. Diese Datensammlung enthält über 17 Millionen Einträge, die Aufschluss darüber geben, welcher Benutzer welche Schlagworte an welche Dokumente zugewiesen hat. Abbildung 34 zeigt einen Auszug aus dem ursprünglichen CiteULike Korpus. Jede Zeile repräsentiert die Zuweisung eines Schlagworts zu einem Dokument durch einen Benutzer zu einem bestimmten Zeitpunkt. Die Daten sind im Format Dokument ID | Benutzer ID | Zeitstempel | Schlagwort abgelegt.

```
4868|808eb76a3ebc6efe03feae67607af389|2004-12-28 19:09:25.48449+00|language
4868|808eb76a3ebc6efe03feae67607af389|2004-12-28 19:09:25.48449+00|object-oriented
4487|cf37904139af49517bbd44deb175adbe|2004-12-28 20:46:48.286988+00|automatic-learning
4487|cf37904139af49517bbd44deb175adbe|2004-12-28 20:46:48.286988+00|google
4487|cf37904139af49517bbd44deb175adbe|2004-12-28 20:46:48.286988+00|linguistics
4487|cf37904139af49517bbd44deb175adbe|2004-12-28 20:46:48.286988+00|ontology
4487|cf37904139af49517bbd44deb175adbe|2004-12-28 20:46:48.286988+00|semantic
24987|808eb76a3ebc6efe03feae67607af389|2004-12-28 21:20:03.627792+00|dirichlet
515|cf37904139af49517bbd44deb175adbe|2004-12-28 21:20:54.146183+00|classification
515|cf37904139af49517bbd44deb175adbe|2004-12-28 21:20:54.146183+00|information-extraction
515|cf37904139af49517bbd44deb175adbe|2004-12-28 21:20:54.146183+00|linguistics
49739|808eb76a3ebc6efe03feae67607af389|2004-12-28 21:30:14.072349+00|relativistic
49739|808eb76a3ebc6efe03feae67607af389|2004-12-28 21:30:14.072349+00|energy
49739|808eb76a3ebc6efe03feae67607af389|2004-12-28 21:30:14.072349+00|momentum
70684|c9bfc05aed3a35f20519a8df5def3721|2004-12-28 21:30:28.653656+00|fixpoint
70684|c9bfc05aed3a35f20519a8df5def3721|2004-12-28 21:30:28.653656+00|haskell
70684|c9bfc05aed3a35f20519a8df5def3721|2004-12-28 21:30:28.653656+00|monads
70684|c9bfc05aed3a35f20519a8df5def3721|2004-12-28 21:30:28.653656+00|recursion
515|808eb76a3ebc6efe03feae67607af389|2004-12-28 22:46:56.6674+00|automatic
515|808eb76a3ebc6efe03feae67607af389|2004-12-28 22:46:56.6674+00|categorization
515|808eb76a3ebc6efe03feae67607af389|2004-12-28 22:46:56.6674+00|classification
515|808eb76a3ebc6efe03feae67607af389|2004-12-28 22:46:56.6674+00|compression
515|808eb76a3ebc6efe03feae67607af389|2004-12-28 22:46:56.6674+00|extraction
515|808eb76a3ebc6efe03feae67607af389|2004-12-28 22:46:56.6674+00|generic
515|808eb76a3ebc6efe03feae67607af389|2004-12-28 22:46:56.6674+00|information
515|808eb76a3ebc6efe03feae67607af389|2004-12-28 22:46:56.6674+00|language
515|808eb76a3ebc6efe03feae67607af389|2004-12-28 22:46:56.6674+00|recognition
515|808eb76a3ebc6efe03feae67607af389|2004-12-28 22:46:56.6674+00|sequences
597|808eb76a3ebc6efe03feae67607af389|2004-12-28 23:28:32.874144+00|finance
597|808eb76a3ebc6efe03feae67607af389|2004-12-28 23:28:32.874144+00|game
597|808eb76a3ebc6efe03feae67607af389|2004-12-28 23:28:32.874144+00|quantum
```

Abbildung 34: Auszug aus dem "Who-posted-what" Korpus von CiteULike im Format Dokument ID | Benutzer ID | Zeitstempel | Schlagwort

[60] http://www.citeulike.org/faq/data.adp

Aus den insgesamt 17 Millionen Zuweisungseinträgen wurden für das ROBUS Testkorpus nur jene übernommen, die sich auf eines der im Testkorpus befindlichen Dokumente (vgl. Kapitel 6.1.2) beziehen. Weiters wurde eine Bereinigung der Schlagwortzuweisungen durchgeführt und im Zuge dessen ein selbstdefiniertes Regelwerk angewandt, um Schlagwörter mit keiner oder sehr geringer inhaltlicher Aussagekraft zu eliminieren. Zu diesen Einträgen zählten alle Schlagwörter, die nur aus einem oder zwei Buchstaben bestanden sowie Wörter, die sich nicht auf den Dokumentinhalt bezogen (zum Beispiel „todo“) oder durch eine externe Anwendung automatisiert vergeben worden waren (zum Beispiel „Autoimport“). Nach Übernahme der Schlagwortzuweisungen und Durchführung der Datenbereinigung enthielt das ROBUS Testkorpus 714.747 Schlagwortzuweisungen von über 18.288 unterschiedlichen Benutzern.

Insgesamt kommen dabei 83.609 unterschiedliche Schlagwörter zum Einsatz. Das ergibt eine durchschnittliche Verwendungshäufigkeit von 8,55 über alle Schlagwortzuweisungen. Die Bandbreite der Verwendungshäufigkeit ist jedoch sehr groß. Zu den am Häufigsten verwendeten Begriffen zählen „internet“ (8.214 Zuweisungen), „database“ (7.891), „review“ (5.473 Zuweisungen) und „bioinformatics“ (3.650 Zuweisungen). Umgekehrt existiert für 42.310 Schlagwörter im Testkorpus nur eine einzige Zuweisung. Die Darstellung in Abbildung 35 zeigt diesen Sachverhalt als Diagramm, wobei auf der y-Achse die Anzahl der Zuweisungen und auf der x-Achse die Anzahl der Schlagwörter, auf die diese Anzahl zutrifft, aufgetragen ist. Die Achsenskalierung ist logarithmisch (Basis 2) ausgeführt. Dabei ist deutlich erkennbar, dass es einige wenige Schlagwörter gibt, die sehr oft zugewiesen werden (linker oberer Diagrammbereich) und viele Schlagwörter, die nur sehr selten verwendet werden (rechter unterer Diagrammbereich).

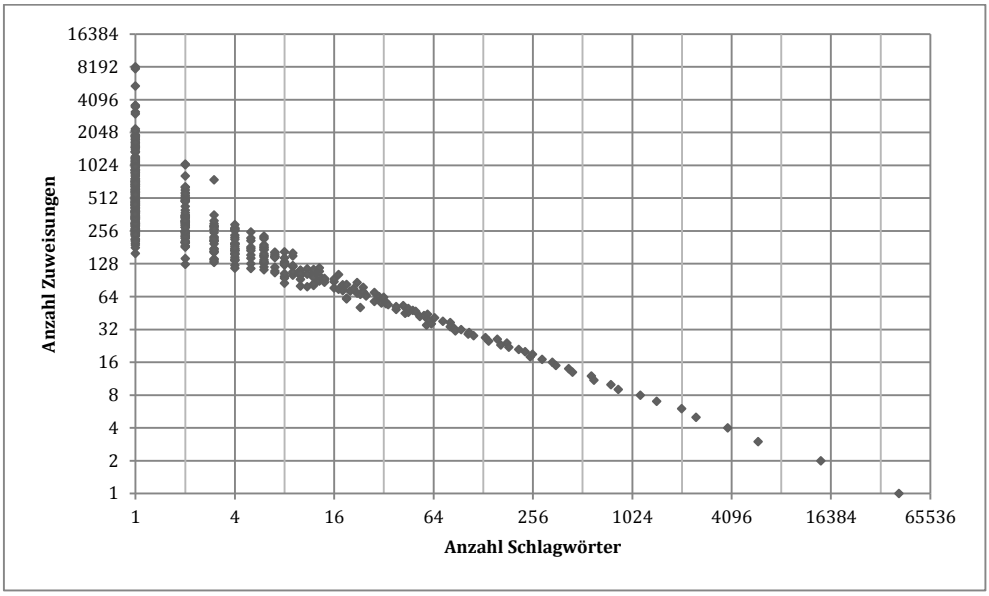

Abbildung 35: Verteilung der Zuweisungshäufigkeit nach Schlagwörtern im ROBUS Testkorpus

Eine vergleichbare Situation ist beim Verhältnis von Benutzer/inne/n zur Anzahl der vergebenen Schlagwörter erkennbar. Das Diagramm in Abbildung 36 zeigt auf der y-Achse die Anzahl der vergebenen Schlagwörter und auf der x-Achse die Anzahl der Benutzer/innen, die diese Menge an Schlagwörtern zugewiesen hat (beide Achsen logarithmisch zur Basis 2 skaliert). Auch hier ist klar ersichtlich, dass eine kleine Anzahl von Benutzer/inne/n viele Schlagwörter verwendet (linker oberer Diagrammbereich), während der Großteil der Benutzer/innen nur eine kleine Menge von Schlagwörtern zugewiesen hat (rechter unterer Diagrammbereich). So gibt es einige Benutzer/innen, die mehrere Tausend Tags zur Beschlagwortung ihrer CiteULike Dokumente verwenden. Hingegen gibt es jeweils ca. 2.500 Benutzer/innen, die insgesamt nur ein oder zwei Schlagwörter eingetragen haben.

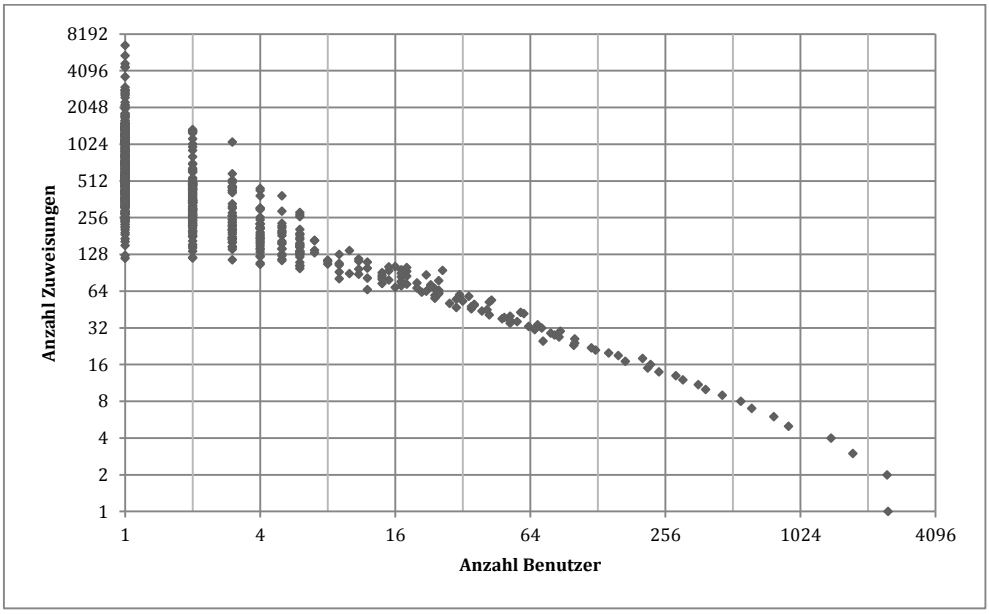

Abbildung 36: Verteilung der Zuweisungshäufigkeit nach Anzahl der Benutzer im ROBUS Testkorpus

Analog zu den beiden vorherigen Darstellungen zeigt Abbildung 37, dass auch die Häufigkeit der zugewiesenen Schlagworte pro Dokument sehr stark variiert. Die y-Achse des Diagramms definiert die Anzahl der zugewiesenen Schlagwörter, die x-Achse die Anzahl der Dokumente, auf die die angegebene Zuweisungshäufigkeit zutrifft (beide Achsen logarithmisch zur Basis 2 skaliert). Während knapp die Hälfte aller Dokumente (33.700) nur mit vier oder weniger Schlagwörtern beschrieben sind (rechter unterer Diagrammbereich), existieren im gesamten Korpus nur zwei Dokumente mit mehr als tausend zugewiesenen Tags (linker oberer Diagrammbereich).

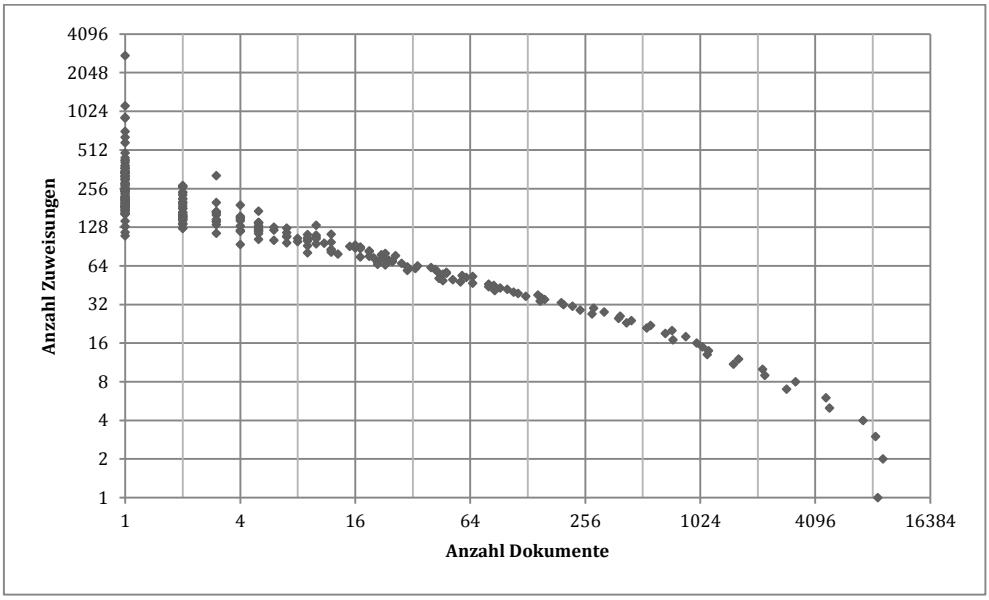

Abbildung 37: Verteilung der Zuweisungshäufigkeit nach Anzahl der Dokumente im ROBUS Testkorpus

Wie schon zuvor die Dokumente, wurden auch alle hier angeführten Daten (Schlagwörter, Benutzer und Zuweisungen) in einer relationalen Datenbank gespeichert. Diese Form der Ablage bietet ein leistungsstarkes und flexibles Grundgerüst und ermöglicht so eine bestmögliche Weiterverarbeitung der Informationen zur Evaluation eines personalisierten Informationssuchsystems. Auf Grundlage dieser Datenstruktur kann die CiteULike Folksonomy entsprechend der Definition von (Vallet et al. 2010) als vierwertiges Tupel $F = (T, U, D, A)$ abgebildet werden:

- $T = (\{t_1, ..., t_m \}$ die Menge aller in der Folksonomy existierenden Tags
- $U = \{ u_1, ..., u_n \}$ die Menge aller aktiven Benutzer
- $D = \{ d_1, ..., d_o \}$ die Menge aller vorhandenen Dokumente
- $A = \{ t_m, u_n, d_o\} \in T \times U \times A$ die Menge aller Zuweisungen eines Tags T zu einem Dokument D durch einen Benutzer U

Die Abbildung der Folksonomy in Form des Tupels ermöglicht es wiederum, die vorhandenen Schlagwörter T eines Benutzers U als dessen individuelle Suchanfragen und die

mit den Tags beschlagworteten Dokumente *D* als die zugehörigen Relevanzbeurteilungen zu interpretieren (Xu et al. 2008). Wenn also ein Benutzer u_1 eine Suchanfrage mit dem Stichwort $q_1 \in T$ formuliert, gelten alle Dokumente d_o des Suchergebnisses als relevant, für die eine Zuweisung $a_{(q1, u1, do)}$ existiert und die folglich in der Teilmenge $A' = \{ q_1, u_1, d_o\} \in T \times U \times A$ vorhanden sind. Die genaue Vorgehensweise und Umsetzung dieses Ansatzes wird in Kapitel 3.6 detailliert erläutert.

6.1.4 Evaluationsmetrik

Wie im vorherigen Kapitel gezeigt, können personalisierte Suchanfragen und Relevanzbeurteilungen automatisiert mit Hilfe von Folksonomies aus Schlagwortzuweisungen generiert werden. Zur Beurteilung der Effektivität eines Informationssuchsystems reicht diese Information alleine aber noch nicht aus. Vielmehr bedarf es einer passenden Evaluationsmetrik, um die Qualität der vom System gelieferten Suchergebnisse und ihrer gereihten Dokumente hinsichtlich der Relevanz der Dokumente für den jeweiligen Benutzer in Bezug auf die Suchanfrage quantitativ zu messen.
Zur Bewertung von gereihten Suchergebnissen im Bereich der Informationssuche existieren in der Literatur zahlreiche Maße und Kennzahlen. Die wichtigsten sind in den Kapiteln 3.8 bis 3.10 ausführlich beschrieben. Zur Evaluation von ROBUS wurde die Mean Average Precision (MAP) als zentrale Kennzahl gewählt. Bei der Ermittlung dieser Kennzahl wird für jede Suchanfrage eines Benutzers die durchschnittliche Genauigkeit (Average Precision) des erhaltenen Suchergebnisses berechnet. Anschließend wird das arithmetische Mittel aller Average Precision Werte gebildet.

$$MAP(Q) = \frac{1}{|Q|} * \sum_{j=1}^{|Q|} \frac{1}{|R|} * \sum_{j=1}^{|R|} P(r_j)$$

Gleichung 11: Berechnung der Mean Average Precision (MAP) als arithmetisches Mittel der Average Precision Werte aller Suchergebnisse für einen Benutzer

Gleichung 11 zeigt die Formel zur Berechnung der Mean Average Precision (MAP) als arithmetisches Mittel der Average Precision Werte aller Suchergebnisse für einen Benutzer, wobei *P* den Precision Wert bei Rang *j*, *R* die Menge aller relevanten Elemente in einem Suchergebnis und *Q* die Menge aller Suchergebnisse für einen bestimmten Benutzer darstellt. MAP wurde als zentrale Bewertungsmetrik zur Evaluation von ROBUS ausgewählt, da es nicht nur eine sehr weit verbreitete und damit aussagekräftige und gut vergleichbare Kennzahl ist, sondern auch, weil das ROBUS Testkorpus ausschließlich binäre Relevanzbeurteilungen („relevant“ oder „nicht relevant“) enthält und für die Beurteilung nicht nur die obersten Ergebnisse berücksichtigt werden sollen, gleichzeitig aber die Reihung innerhalb der ersten k Ergebnisse sehr wohl ausschlaggebend ist. In Anbetracht all dieser Kriterien ist MAP aufgrund ihrer Robustheit und Aussagekraft die am besten geeignete Kennzahl zur Bewertung der Effektivität des ROBUS Systems.
Wie oben erläutert, bezieht sich jeder MAP Wert auf den Mittelwert aller Suchergebnisse eines Benutzers. Da es sich bei ROBUS aber um ein personalisiertes Suchsystem handelt, ist es auch bei der Testdurchführung nötig, die Suchanfragen und Suchergebnisse in Abhängigkeit von verschiedenen Benutzerprofilen zu evaluieren. Dies würde jedoch bedeuten, dass das Bewertungsergebnis wiederum nicht als eine aggregierte Kennzahl, sondern als eine Menge von einzelnen MAP Werten vorliegen würde. Daher wird die MAP Metrik folgend dem von (Xu et al. 2008) vorgestellten Ansatz erweitert. Dabei wird die Mean MAP (MMAP) Kennzahl als arithmetischer Mittelwert aller MAP Werte für alle Benutzer berechnet.

$$MMAP(U) = \frac{1}{|U|} * \sum_{u=1}^{|U|} MAP(u)$$

Gleichung 12: Berechnung der Mean MAP (MMAP) Kennzahl als arithmetisches Mittel aller MAP Werte aller Benutzer

Obige Gleichung zeigt die Formel zur Berechnung der MMAP Kennzahl nach (Xu et al. 2008), wobei *U* der Menge aller evaluierten Benutzer und *MAP(u)* dem MAP Wert aller Suchergebnisses eines Benutzers *u* entspricht.

6.1.5 Erstellung von Evaluationsrollenprofilen

Das ROBUS Testkorpus enthält neben einer Sammlung von 77.000 akademischen Artikeln noch eine Menge von ca. 715.000 individuellen Relevanzbeurteilungen in Form von Schlagwortzuweisungen aus CiteULike. Obwohl das Korpus damit alle wesentlichen Merkmale einer Datensammlung zur Evaluation eines personalisierten Suchsystems aufweist, fehlt noch eine essentielle Komponente, ohne die eine Effektivitätsbewertung von ROBUS nicht möglich ist. Wie in Kapitel 0 ausführlich erläutert, verwendet das ROBUS System einen kontext-sensitiven Algorithmus, der unter Zuhilfenahme von Rollenprofilen die ursprüngliche Reihung der Suchergebnisse so anpasst, dass jene Elemente an oberster Stelle des Suchergebnisses erscheinen, die die höchste Relevanz für das Rollenprofil, das dem suchenden Benutzer zugeordnet ist, aufweisen (Details siehe Kapitel 5.1). Das ROBUS Testkorpus enthält zwar ca. 18.000 unterschiedliche Benutzer und die Schlagwörter, die von ihnen in CiteULike zugeordnet wurden. Es existieren aber keine Rollendefinitionen beziehungsweise –zuordnungen. Ohne diese Informationen ist jedoch eine Suche mit ROBUS - und dementsprechend auch die Evaluation des Systems - unmöglich, da der oben beschriebene Algorithmus diese Rollendefinitionen und –zuordnungen zwingend benötigt. Diesem Problem Abhilfe schaffen, könnte das „Group Membership Dataset“, das auf der CiteULike Dataset Website[61] für registrierte BenutzerInnen und akademische Zwecke kostenlos verfügbar ist. Es steht in Form einer einfachen Textdatei mit dem Format „Gruppenkennzeichner“ | „Benutzerkennzeichner“ zum Download

[61] http://www.citeulike.org/faq/data.adp

bereit. Jede Zeile repräsentiert dabei die Zugehörigkeit eines Benutzers zu einer CiteULike Gruppe. Jeder Benutzer kann beliebig vielen Gruppen angehören. Für jede Mitgliedschaft des Benutzers wird eine neue Zeile hinzugefügt.

Jeder Gruppe dieser Datensammlung könnte für die Evaluation von ROBUS als eine Unternehmensrolle und jede Gruppenmitgliedschaft als Zuordnung eines Mitarbeiters (Benutzer) zu einer Rolle (Gruppe) interpretiert werden. Obwohl dieser Ansatz auf den ersten Blick sehr vielversprechend wirkt, kann er nicht zur Erstellung von Evaluationsprofilen für ROBUS verwendet werden. Die Gründe dafür liegen einerseits in der verhältnismäßig kleinen Anzahl an vorhandenen Gruppenzugehörigkeiten (der Anteil an CiteULike Benutzern, die zumindest einer Gruppe angehören, liegt bei < 10 Prozent) und andererseits darin, dass über die Gruppe keine Information außer dem Kennzeichnungsfeld vorhanden ist. Dieses wird jedoch nur durch einen abstrakten MD5[62] Schlüssel beschrieben (zum Beispiel „d21d7cx98a02b903e9440956ede8427g“), der keinerlei semantische Aussagekraft besitzt und damit nicht als Rollenbezeichnungskriterium für den ROBUS Profilerstellungsalgorithmus (vgl. Kapitel 5.1.3) verwendet werden kann.

Da seitens CiteULike keine expliziten Rolleninformationen zur Generierung von Rollenprofilen existieren, wurden diese Daten für die Evaluation von ROBUS aus den verfügbaren Schlagwortzuweisungen konstruiert. Das Auswerten von Folksonomy Daten im Allgemeinen sowie das Extrahieren von Benutzerprofilen aus Social Tagging Daten im Speziellen hat in der jüngsten Vergangenheit vermehrt Aufmerksam in der wissenschaftlichen Community auf sich gezogen. Die Arbeit von Bouadjenek et al. gibt einen guten Überblick über verschiedene aktuelle Arbeiten auf diesem Gebiet und enthält darüber hinaus auch detaillierte Evaluationsergebnisse zu den präsentierten Methoden. (Bouadjenek et al. 2013)

Ein konkretes Beispiel für die Auswertung von Social Tagging Informationen zur Personalisierung von Suchsystemen ist die Arbeit von (Vallet et al. 2010). Die Autoren präsentieren darin eine Vorgehensweise, bei der die jeweils populärsten (am häufigsten verwendeten) Tags eines Benutzer einerseits zur Generierung von Benutzerprofilen und anderer-

[62] MD5 ... Message-Digest Algorithm 5; ursprünglich von Robert Rivest 1991 als kryptografische Hashfunktion entwickelt: http://tools.ietf.org/html/rfc1321

seits zur Bestimmung von personalisierten Informationsbedürfnissen (Topics) herangezogen werden. Die auf diese Weise generierten Topics werden in weiterer Folge als Suchanfrage in Form einzelner Stichwörter an das zu evaluierende System übermittelt.
Die von (Vallet et al. 2010) vorgestellte Methode wurde in angepasster Form auch für die Erzeugung von Rollenprofilen zur Evaluation von ROBUS verwendet. Analog zur oben beschriebenen Methode von Vallet et al. wurden dabei ebenfalls die populärsten Schlagwörter der Benutzer aus CiteULike selektiert und diese als textuelle Rollenbezeichnungen interpretiert. Mit Hilfe dieser Vorgehensweise kann einerseits jeder Benutzer einer Rolle zugeordnet werden und andererseits können die den Rollenprofilen zugrunde liegenden gewichteten Termvektoren anhand des in Kapitel 5.1 beschriebenen Verfahrens automatisiert erstellt werden.

6.1.6 Selektion einer kompetitiven Baseline

Um eine wirklich aussagekräftige Beurteilung der Effektivität eines personalisierten Informationssuchsystems abgeben zu können, ist es notwendig, die Testergebnisse dieses Systems einem Vergleichssystem (engl. Baseline) gegenüber zu stellen. Nur so kann die tatsächliche Verbesserung einer neuen Methode im Vergleich zu aktuellen Technologien festgestellt und der Fortschritt der Forschungsbemühungen langfristig sichergestellt werden (Armstrong et al. 2009).
Folglich bedarf es zur Evaluation von ROBUS neben den Testdokumenten (vgl. Kapitel 6.1.2), den personalisierten Suchanfragen und Relevanzbeurteilungen (vgl. Kapitel 6.1.3), einer definierten Evaluationsmetrik (vgl. Kapitel 6.1.4) und den automatisiert erstellten Rollenprofilen (vgl. Kapitel 6.1.5) noch einem separaten Informationssuchsystem, das als Baseline fungieren kann. Bei der Auswahl eines solchen Systems sollte darauf geachtet werden, dass es die folgenden Kriterien erfüllt:

- **Kompetitiv.** Das Vergleichssystem ist innerhalb der wissenschaftlichen Forschung als „State-of-the-Art“ (englisch für: auf dem neuesten Stand der Technik) System anerkannt und gilt somit als kompetitives Vergleichsmaß. Die Auswahl

eines potentiell veralteten beziehungsweise „schwachen“ Vergleichssystems kann die relative Verbesserung des evaluierten Systems erhöhen, stellt aber eine wesentliche Verzerrung des Evaluationsergebnisses dar und ist somit unbedingt zu vermeiden.

- **Transparent.** Die grundlegenden Methoden, Algorithmen und Technologien, auf denen das Vergleichssystem basiert, sollten dokumentiert und somit das Verhalten und die Ergebnisse des Systems nachvollziehbar sein. Die Darstellung einer verbesserten Sucheffektivität des eigenen Systems gegenüber einem „Black Box“ System, d.h. einem System, dessen interner Aufbau und interne Arbeitsweise nicht bekannt ist und dementsprechend nicht nachvollzogen werden kann, ist als problematisch zu betrachten und sollte vermieden werden.
- **Reproduzierbar.** Die verwendeten Testergebnisse eines Baseline Systems sollten zu einem späteren Zeitpunkt beziehungsweise von anderen Forschungsgruppen reproduziert werden können, damit eine höchstmögliche Nachvollziehbarkeit der Testergebnisse gegeben ist. Dies kann entweder dadurch erreicht werden, indem die konkrete Implementierung des verwendeten Baseline Systems für andere Forschungsgruppen verfügbar ist (zum Beispiel in Form eines Open Source[63] Systems), oder indem das verwendete System ausreichend detailliert beschrieben wird, um von anderen Forschern zu einem späteren Zeitpunkt nachgebildet werden zu können (vergleiche auch Abschnitt „Transparent“). In jedem Fall ist aber darauf zu achten, dass auch gegebenenfalls verwendete relevante Konfigurationseinstellungen (zum Beispiel Gewichtungsparameter) dokumentiert werden. Als relevant in diesem Sinne gelten dabei alle Parameter, die einen direkten oder indirekten Einfluss auf die evaluierten Kenngrößen aufweisen.
- **Vergleichbar.** Testergebnisse von unterschiedlichen Systemen sollten nur dann miteinander verglichen werden, wenn sie auch unter vergleichbaren Rahmenbedingungen zustande gekommen sind. Wie bereits in den vorherigen Kapiteln erläutert, hat jede einzelne bei der Evaluation verwendete Komponente (Testdoku-

[63] „The Open Source Definition“: http://opensource.org/docs/osd

mente, Suchanfragen, Relevanzbeurteilungen, usw.) großen Einfluss auf die Qualität der gelieferten Suchergebnisse und somit auf die Testergebnisse im Gesamten.

Obwohl die hier genannten Kriterien innerhalb der Forschung hinlänglich bekannt und weitgehend akzeptiert sind, werden sie trotzdem häufig missachtet. So haben beispielsweise Armstrong et al. im Zuge einer selbst durchgeführten Studie die Testmethoden und -ergebnisse mehrerer TREC Tracks, die im Zuge der SIGIR[64] Konferenzreihe in den Jahren 1998 bis 2008 und im Zuge der CIKM[65] Konferenzreihe zwischen 2004 und 2008 veröffentlicht wurden, analysiert. Dabei haben sie festgestellt, dass die von den Forschern verwendeten Baseline Systeme „im Allgemeinen sehr schwach" sind, und dass viele davon bei genauerer Betrachtung nicht einmal mit den Ergebnissen des ursprünglichen TREC Systems mithalten können.
Sie kritisieren des Weiteren, dass alleine bei den für TREC-8 getesteten Systemen mehr als die Hälfte der Autoren ihre Ergebnisse mit Baseline Werten vergleichen, die weniger effektiv, als der Durchschnitt der automatisierten TREC Testtools aus dem Jahre 1999 sind. Armstrong et al. kommen zu der erschütternden Erkenntnis, dass im untersuchten Zeitraum keine gesamtheitliche Verbesserung im Bereich der Informationssuche durch die veröffentlichten Methoden festgestellt werden kann. Zur Lösung des Problems schlagen sie den Aufbau eines zentralen und für Forschungszwecke frei zugänglichen Verzeichnisses mit verfügbaren Baseline Systemen und Referenzergebnissen vor (Armstrong et al. 2009).

---

64 ACM SIGIR ... Association for Computing Machinery's Special Interest Group on Information Retrieval: http://www.sigir.org

65 CIKM ... Conference on Information and Knowledge Management: http://www.cikmconference.org

6.1.7 BM25 als Baseline für ROBUS

In Anbetracht der im vorigen Kapitel beschriebenen Kriterien und unter Berücksichtigung der oben angeführten Ausgangssituation wurde als Baseline für die ROBUS Evaluation ein auf dem BM25 Algorithmus basierendes Suchsystem gewählt. BM25 wird heute von verschiedenen Suchmaschinen als Funktion zur Reihung von Suchergebnissen verwendet und gilt innerhalb der wissenschaftlichen Forschung im Bereich der Informationssuche als die State-of-the-Art Methode (Perez-Iglesias et al. 2009; Trotman & Keeler 2011; Garrido et al. 2010; Clinchant 2012; Robertson & Zaragoza 2009; Robertson 1997). In der Literatur findet sich häufig auch die Bezeichnung Okapi BM25, was darauf zurückzuführen ist, dass das Okapi[66] Informationssuchsystem das erste war, das diese Funktion implementierte. (Robertson 1997)

Der BM25 Algorithmus basiert auf dem Konzept des „Probabilistic Relevance Framework“ (PRF), welches bereits in den 1970er und 1980er Jahren entwickelt wurde. Das PRF Modell ermittelt eine Wahrscheinlichkeit, die darüber Auskunft gibt, ob beziehungsweise wie ein Dokument *d* für eine Suchanfrage *q* relevant ist. Dieser Wahrscheinlichkeitsfunktion liegt die Annahme zugrunde, dass eine Teilmenge an Dokumenten *R* aus der Menge aller Dokumente *D* vom Benutzer als relevant in Bezug auf eine bestimmte Suchanfrage erachtet und somit als Suchergebnis zurück geliefert werden sollte. Ausgehend von diesen grundsätzlichen Überlegungen wurden von Robertson et al. mehrere konkrete Implementierungen des Modells, wie etwa das „Binary Independence Model“, das „Eliteness Model“ oder das „2-Poisson Model“ entwickelt. Das am weitetesten verbreitete Modell ist jedoch ohne Zweifel die „BM25 Term-weighting and Document-scoring Function“ (Robertson & Zaragoza 2009). Für eine detaillierte Beschreibung der Funktion sowie der ihr zugrunde liegenden Theorie sei an dieser Stelle auf Kapitel 2.4 verwiesen.

Als konkrete Implementierung der BM25 Gewichtungs- und Reihungsfunktion wurde die Umsetzung von (Perez-Iglesias et al. 2009) verwendet. Sie wurde als Erweiterung des

[66] Okapi ... Familie von Informationssuchsystemen, die in den 1990-er Jahren unter der Leitung von Stephen Robertson am University College London entwickelt wurden: http://www.soi.city.ac.uk/~ser/

Open Source Suchsystems „Apache Lucene“ konzipiert und ist seit Version 4.0 Teil des Standardfunktionsumfangs der Applikation. (Foundation 2012)
Die Implementierung von Perez-Iglesias et al. verwendet die folgenden Gleichungen, um die BM25 Funktion in Apache Lucene abzubilden:

$$R(q,d) = \sum_{t \in q} idf(t) * \frac{occurs_t^d}{k_1 * \left((1-b) + b * \frac{l_d}{avl_d}\right) + occurs_t^d}$$

Gleichung 13: Berechnung des BM25 Gewichtes von *d* für eine Suchanfrage *q* in Apache Lucene nach (Perez-Iglesias et al. 2009)

Die oben dargestellte Formel in Gleichung 13 beschreibt die Berechnung der BM25 Gewichtung eines Dokuments *d* in Bezug auf eine Suchanfrage *q*, wobei $occurs_t^d$ die Vorkommenshäufigkeit (Termfrequenz, englisch: Term Frequency) des Terms *t* in einem Dokument *d* widerspiegelt. Die Länge des aktuellen Dokuments ist mit l_d gekennzeichnet, während avl_d die durchschnittliche Länge aller Dokumente in der Datensammlung angibt. Des Weiteren finden sich in der Formel der Parameter k_1, dessen Wert prinzipiell frei gewählt werden kann sowie der Parameter *b*, dessen Wert sich im Bereich $b \in [0,1]$ befinden muss. Als Standardkonfiguration für die beiden Parameter werden die Werte $k_1 = 2$ und $b = 0{,}75$ definiert. Die Berechnung von *idf(t)* (Inverse Dokumentfrequenz, englisch: Inverse Document Frequency) ist in der untenstehenden Formel festgehalten.

$$idf(t) = \log\left(\frac{N - df(t) + 0{,}5}{df(t) + 0{,}5}\right)$$

Gleichung 14: Berechnung der inversen Dokumentfrequenz (idf) für einen Term *t* in Apache Lucene nach (Perez-Iglesias et al. 2009)

Der *idf*-Wert wird für jeden in der Datensammlung vorkommenden Term *t* entsprechend der obigen Formel in Abhängigkeit von *N* (Anzahl aller Dokumente in der Datensammlung) und *df* (Anzahl der Dokumente, in denen Term *t* mindestens ein Mal vorkommt) ermittelt.
Apache Lucene ist ein umfangreiches Informationssuchsystem, das zur Indexierung und Suche in großen natürlichsprachlichen Textsammlungen entwickelt wurde. Es zählt heute zu einem der populärsten und am weitesten verbreiteten Systeme dieser Art, sowohl im wissenschaftlichen als auch im kommerziellen Umfeld. Die Implementierung erfolgte ursprünglich mittels der Programmiersprache Java[67], mittlerweile gibt es aber auch Portierungen für andere Programmiersprachen wie beispielsweise .Net[68] oder Python[69].
Apache Lucene ist ein Open Source Projekt, dessen veröffentlichte Implementierungsversionen von der Lucene Website[70] heruntergeladen und im Rahmen der Apache Software License[71] kostenlos genutzt werden können. In der Standardkonfiguration verwendet das System eine Gewichtungs- und Reihungsfunktion, die auf dem Konzept des „Vector Space Model“ (vgl. Kapitel 2.1) basiert. Das freie und offene Implementierungskonzept erlaubt aber auch den Einsatz von anderen Algorithmen wie eben beispielsweise BM25 (Perez-Iglesias et al. 2009).

Für die Evaluation von ROBUS wurde Apache Lucene in der Version 4.0 mit der oben beschriebenen Implementierung der BM25 Funktion als Basline System gewählt, da diese Konfiguration nicht nur zur Zeit als State-of-the-Art in der Informationssuche betrachtet werden kann, sondern darüber hinaus auch auf eine langjährige und langfristig erfolgreiche Entwicklung zurückblicken kann. Dies ist in der Literatur ausführlich belegt und wird

---

[67] Java ist eine sehr weit verbreitete Programmiersprache, die ursprünglich von Sun Microsystems im Jahre 1995 vorgestellt wurde: http://www.oracle.com/technetwork/java/index.html

[68] .Net ist eine Software-Plattform der Firma Microsoft zur Entwicklung und Ausführung von Anwendungen: http://www.microsoft.com/net

[69] Python ist eine frei verfügbare und betriebssystemunabhängige Programmiersprache, die hauptsächlich zur Entwicklung von Webapplikationen verwendet wird: http://www.python.org/

[70] http://lucene.apache.org/core/

[71] Apache Software License: http://www.apache.org/licenses/

zusätzlich von der großen Anzahl an Benutzer/inne/n untermauert. Darüber hinaus basieren alle wesentlichen verwendeten Komponenten des Systems auf wissenschaftlich fundierten Modellen, die detailliert dokumentiert und damit nachvollziehbar sind. Ein weiterer ausschlaggebender Grund für die Selektion von Apache Lucene und BM25 als Baseline bei der ROBUS Evaluation war die freie Verfügbarkeit (gesamter Source Code kann kostenlos herunter geladen werden) und die damit gewährleistete Reproduzierbarkeit (vgl. Kapitel 6.1.6) der Teststellung sowie die hohe Akzeptanz des Systems innerhalb der wissenschaftlichen Community.

6.2 Evaluationsergebnisse für ROBUS

6.2.1 Konkrete Testkonfiguration

Zur konkreten Durchführung der Effektivitätsbewertung von ROBUS wurden auf Grundlage des oben beschriebenen Evaluationskorpus drei unterschiedliche Testsätze mit unterschiedlichen Benutzern, Dokumenten und Schlagwortzuweisungen generiert. Die Auswahl der Daten erfolgte anhand der Anzahl von Schlagworten, die ein/e Benutzer/in insgesamt zugewiesen hat. Der erste Datensatz *CUL100* enthielt die Dokumente und Schlagwortzuweisungen von insgesamt 100 Benutzer/inne/n, die jeweils circa 100 Schlagworte (zwischen 90 und 110) vergeben hatten. Analog dazu, wurden für den Testsatz *CUL500* die Dokument- und Schlagwortdaten von 100 Benutzer/inne/n mit jeweils circa 500 Zuweisungen (zwischen 450 und 550) ausgewählt. Der dritte Testsatz enthielt die Daten von 100 Benutzer/inne/n, die jeweils ungefähr 1500 Schlagworte (zwischen 1400 und 1600) verwendeten und wurde dementsprechend mit *CUL1500* bezeichnet.

Die Aufteilung eines Testkorpus in unterschiedlich große Datensätze wird bei der Evaluation von Informationssuchsystemen sehr häufig angewendet (siehe beispielsweise (Harpale et al. 2010; Xu et al. 2008; Vallet et al. 2010)), da dadurch variierende Testergebnisse in Abhängigkeit der zugrunde liegenden Mengengerüste erkannt und analysiert werden können.

Anschließend wurde für jede/n der in einem der drei Testsätze enthaltenen Benutzer/innen ein Rollenprofil erstellt und zugewiesen. Die Erstellung der Rollenprofile erfolgte mittels der in Kapitel 6.1.5 erläuterten Vorgehensweise, wobei die Anzahl der im Rollenvektor verwendeten Terme mit 20 festgelegt wurde. Des Weiteren wurden für jede/n Benutzer/in 25 Suchanfragen und dazugehörige Relevanzbeurteilungen folgend der Logik in Kapitel 6.1.3 aus den Schlagwortzuweisungsdaten extrahiert. Das grundlegende Prinzip dabei ist, dass ein Benutzer, der nach einem Begriff sucht, den er auch als Schlagwort (Tag) Dokumenten zugewiesen hat, diese Dokumente als relevant in Bezug auf die Suchanfrage erachtet und dementsprechend diese Dokumente im Suchergebnis wiederfinden möchte.

Folgend diesem Prinzip wird ein Dokument im Suchergebnis von der ROBUS Evaluationsmetrik (vgl. Kapitel 6.1.4) dann als relevant erachtet, wenn der suchende Benutzer das Dokument mit dem Tag beschlagwortet hat. Voraussetzung dafür ist natürlich, dass sich die formulierte Suchanfrage in den Schlagwörtern des Benutzers wiederfindet. Die Evaluationsmetrik aggregiert die einzelnen Relevanzbewertungen aller Suchanfragen eines Benutzers zum Mean Average Precision (MAP) Wert und in weiterer Folge die MAP Werte aller Benutzer zum Mean MAP (MMAP).

6.2.2 Durchführung und Ergebnisse

Die Suchanfragen der Benutzer/innen aus den drei Testsätzen wurden einerseits an das ROBUS und andererseits an das Baseline (vgl. Kapitel 6.1.7) System übermittelt. Dabei ist wichtig zu erwähnen, dass für beide Systeme stets die identen Datensätze zum Einsatz gekommen sind. Der aggregierte MMAP Wert wurde für jeden Testsatz und jedes System separat berechnet und ausgewertet. Durch diese Vorgehensweise können die Effektivitätsunterschiede der beiden Suchsysteme direkt miteinander verglichen und bewertet werden.

Die Werte in Tabelle 12 zeigen die konkreten Evaluationsergebnisse des Baseline und des ROBUS Systems in Form der aggregierten MMAP Kennzahlen für jeden Testsatz. Ferner zeigen sie, dass das ROBUS System bei allen drei Testsätzen bessere MMAP Werte als das Baseline System erzielt. Die relative Verbesserung des ROBUS Systems gegenüber der Baseline ist in der letzten Zeile der Tabelle als prozentualer Wert angeführt.

Testsatz	**CUL100**	**CUL500**	**CUL1500**
Anzahl Benutzer	100	100	100
Anzahl Terme pro Rollenvektor	20	20	20
Anzahl Suchanfragen pro Benutzer	25	25	25
Maximale Anzahl Elemente im Suchergebnis	75	75	75
Baseline MMAP	0,1005	0,0977	0,1406
ROBUS MMAP	0,1125	0,1100	0,1540
Verbesserung	**11,9%**	**12,6%**	**9,5%**

Tabelle 12: Verbesserung der Sucheffektivität durch ROBUS

Wie der obigen Tabelle entnommen werden kann, ist es durch die Berücksichtigung des langfristigen Benutzerkontexts in Form von Unternehmensrollen bei der Suchstrategie von ROBUS möglich, trotz des äußerst kompetitiven BM25 Vergleichssystems noch zu einer Verbesserung der Sucheffektivität beizutragen.

6.3 Zusammenfassung

Im Kapitel 0 „

Evaluationsmethode & Ergebnisse“ wurden einerseits alle wesentlichen Voraussetzungen, Komponenten und Grundlagen dokumentiert, die zur Evaluation eines personalisierten Informationssuchsystems wie ROBUS erforderlich sind und andererseits auch die eigentlichen Testergebnisse, die zur Beurteilung des Effektivität des ROBUS Systems herangezogen wurden, präsentiert. Das Kapitel enthält eine Beschreibung von wichtigen allgemeinen Evaluationsmethoden. Darüber hinaus wird gezeigt, wie die für jede Evaluation benötigten Suchanfragen und Relevanzbeurteilungen gesammelt beziehungsweise generiert werden können.

Da es sich bei ROBUS um ein personalisiertes Suchsystem handelt, werden die Herausforderungen bei der Evaluation solcher Systeme im Speziellen beleuchtet (vgl. Kapitel 3.5). Insbesondere werden aktuelle Methoden und Systeme zur automatisierten Extraktion von individuellen Suchanfragen und Relevanzbeurteilungen präsentiert. Des Weiteren werden das Konzept der Effektivität zur Beurteilung der Qualität eines Suchsystems vorgestellt und verschiedene Evaluationsmetriken zur quantitativen Bewertung dieses Konzepts präsentiert (vgl. Kapitel 3.8 bis 3.10).

Im zweiten Abschnitt des Kapitels werden die spezifischen Komponenten, Algorithmen und Datensammlungen vorgestellt, die für die Evaluation von ROBUS benötigt beziehungsweise eingesetzt werden. Es wird gezeigt, welche Kriterien bei der Auswahl einer Testdatensammlung relevant sind (vgl. Kapitel 6.1.1) und wie die Daten der Social Bookmarking Website CiteULike verwendet werden können, um ein Evaluationskorpus zu generieren. Außerdem wird das MMAP Konzept zur Bewertung von Suchergebnissen als konkrete Evaluationsmetrik (vgl. Kapitel 6.1.4) sowie eine spezielle Vorgehensweise zur automatisierten Generierung von Rollenprofilen (vgl. Kapitel 6.1.5) für die ROBUS Evaluation dargestellt. Ein weiterer wesentlicher Punkt der Evaluationsmethode für ROBUS wird in Kapitel 6.1.6 erläutert. Es beschreibt die Notwendigkeit eines Vergleichssystems (Baseline) sowie die wesentlichen Kriterien, die bei der Auswahl eines solchen Systems beachtet werden sollten. Der Abschnitt schließt mit der Diskussion zu dem gewählten Baseline System auf Basis einer BM25 Implementierung für das Suchsystem Apache Lucene (vgl. Kapitel 6.1.7).

Der finale Teil des Kapitels beschreibt letztendlich die Konfiguration der relevanten Systemparameter sowie die Erstellung von drei verschiedenen Testsätzen, mit unterschiedlichen Mengengerüsten, die zur Evaluation eingesetzt werden (vgl. Kapitel 6.2.1). Des

Weitern wird die konkrete Vorgehensweise bei der Testdurchführung präsentiert und die Methodik der Relevanzbewertung von Suchergebnissen der beiden getesteten Systeme (ROBUS und Baseline) erläutert.
Die eigentlichen Testergebnisse der beiden Systeme sowie eine Gegenüberstellung dieser wird anschließend in Tabelle 12: Verbesserung der Sucheffektivität durch ROBUS festgehalten. Es wird anhand dieser gezeigt, dass ROBUS für alle analysierten Testsätze bessere Effektivitätskennzahlen erzielt und somit zu einer Verbesserung der Suche beitragen kann (vgl. Kapitel 6.2.2).

7 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wird gezeigt, wie dem Problem der vielfach überbordenden und rasant wachsenden Menge an unstrukturierten Daten in Unternehmen entgegengetreten werden kann. Dabei werden die langfristigen Informationsbedürfnisse der Mitarbeiter/innen in Form ihrer jeweiligen Rolle im Unternehmen mittels eines speziell entwickelten Verfahrens in Relation zu den textuellen (unstrukturierten) Daten gesetzt. Die Relation zwischen Unternehmensrolle und Dokumentinhalt gibt Auskunft darüber, wie relevant ein Dokument für den/die Mitarbeiter/in einer bestimmten Rolle ist.

Auf Grundlage dieses wesentlichen Zusammenhangs wurde im Rahmen der gegenständlichen Arbeit das kontextsensitive rollenbasierte Unternehmenssuchsystem ROBUS entwickelt und evaluiert. ROBUS ist in der Lage für jede definierte Unternehmensrolle ein Rollenprofil in Form eines gewichteten Termvektors zu generieren. Dazu durchforstet das System eine Vielzahl von natürlichsprachlichen Stellenausschreibungstexten und extrahiert jene Begriffe, die die gegebene Unternehmensrolle am besten charakterisieren. Ermöglicht wird dies durch den Einsatz spezieller computerlinguistischer Methoden, mit Hilfe derer die Stellenausschreibungstexte verarbeitet und analysiert werden können. Darüber hinaus wird ein standardisierter, domänenspezifischer Thesaurus eingesetzt, um berufsbezogene Fachbegriffe erkennen und interpretieren zu können. Eine eigens für ROBUS entwickelte Gewichtungsfunktion bewertet die aus den Ausschreibungstexten extrahierten Begriffe, wobei das zugewiesene Gewicht umso höher ist, desto charakterisierter der Begriff für die jeweilige Rolle ist (weitere Details siehe Kapitel 5.1).

In weiterer Folge ermittelt ROBUS für jedes vorhandene Rollenprofil (sprich: für jeden Termvektor) die Relevanz jedes Dokuments in der Dokumentensammlung des Unternehmens für die jeweilige Rolle. Die Bestimmung der Relevanz erfolgt mittels der in Kapitel 2.3 erläuterten Ähnlichkeitsmaße und basiert auf der grundlegenden Annahme, dass die Relevanz eines Dokuments für eine/n Mitarbeiter/in umso höher ist, desto ähnlicher der Termvektor des zugeordneten Rollenprofils einem Dokument ist. Auf Grundlage dieser Annahme werden die Suchergebnisse von ROBUS anhand der Relevanzwerte neu gereiht. Die vollständige Beschreibung des rollensensitiven Suchverfahrens von ROBUS findet sich in Kapitel 5.2.

Ein besonderer Schwerpunkt dieser Arbeit lag auf dem Bereich der Evaluation von kontextsensitiven Suchsystemen im Allgemeinen und ROBUS im Speziellen. Um eine möglichst aussagekräftige Beurteilung der Suchergebnisse gewährleisten zu können, erfolgte eine tiefgehende Analyse von aktuellen Evaluationsmethoden für personalisierte Suchsysteme (siehe Kapitel 0) sowie eine darauf aufbauende Konzeption und Umsetzung eines Testkorpus für rollensensitive Suchsysteme (siehe Kapitel 6.1).

Mit Hilfe der daraus gewonnen Testergebnisse konnte nachweislich belegt werden, dass das rollensensitive Suchsystem ROBUS zu einer deutlichen Verbesserung bei der Suche in unstrukturierten Unternehmensdaten im Vergleich zu nicht-rollensensitiven Systemen führt. Die detaillierten Testergebnisse werden in Kapitel 6.2 präsentiert und diskutiert.

Die vielversprechenden Testergebnisse legen nahe, dass weiterführende Forschungsmaßnahmen hinsichtlich der Optimierung des Systems in mehrere Richtungen zu einer zusätzlichen Ergebnisverbesserung führen würden. In der Phase der automatisierten Rollenprofilgenerierung betrifft dies vor allem die Berücksichtigung von Multitokens, die Bigram-Filterung sowie eine Kollokationsanalyse der zugrunde liegenden Stellenausschreibungstexte. Dadurch könnte die Identifikation von (zusammengesetzten) Fachbegriffen und Eigennamen verbessert werden. Auch die Funktionen zum Abgleich mit dem DISCO Thesaurus (vgl. Kapitel 5.1.5) würden dadurch höhere Matching-Werte erzielen und so zu einer Gesamtverbesserung der Profilvektoren beitragen. Neben der eigentlichen computerlinguistischen Analyse spielt die Gewichtungsfunktion zur Bildung der Termvektoren (vgl. Kapitel 5.1.6) eine wesentliche Rolle. Hierbei kann in künftigen For-

schungsaktivitäten untersucht werden, inwiefern eine Adaption der Funktion (z.B. Optimierung der Gewichtungsparameter) zu einer Verbesserung der Gesamtergebnisse führen kann.

Diese Arbeit konzentrierte sich nur auf ein Einsatzgebiet der generierten Profilvektoren: die rollensensitive Optimierung von Suchfragen in Unternehmensinformationssystemen. Grundsätzlich können die Profilvektoren jedoch überall dort eingesetzt werden, wo Unternehmensrollen mit textuellen Inhalten in Relation gesetzt werden sollen. Ziel von zukünftigen Forschungsaktivitäten könnte es daher sein, auch andere aussichtsreiche Anwendungsmöglichkeiten, wie z.B. die Generierung von rollenbasierten Tag-Clouds für Wissensdatenbanken zu untersuchen.

Literaturverzeichnis

Agichtein, E. et al., 2006. Learning user interaction models for predicting web search result preferences. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp.3–10.

Agichtein, E., Brill, E. & Dumais, S., 2006. Improving web search ranking by incorporating user behavior information. In *29th annual international ACM SIGIR conference on Research and development in information retrieval*. Seattle, Washington, USA: ACM, pp. 19–26.

Anon, 2006. *Duden -- Deutsches Universalwörterbuch*, Mannheim: Bibliographisches Institut.

Anon, 1991. Roget's Thesaurus, 1911. , 2013(22.09). Available at: http://machaut.uchicago.edu/rogets.

Apache-OpenNLP-Development-Community, 2013a. Apache OpenNLP Developer Documentation. Available at: http://opennlp.apache.org/documentation/1.5.3/manual/opennlp.html.

Apache-OpenNLP-Development-Community, 2013b. OpenNLP Developer Documentation - Part-of-Speech Tagger. , 2012(29.05). Available at: http://opennlp.apache.org/documentation/1.5.2-incubating/manual/opennlp.html#tools.postagger.

Apache-OpenNLP-Development-Community, 2013c. OpenNLP Developer Documentation - Tokenizer. , 2012(29.05). Available at: http://opennlp.apache.org/documentation/1.5.2-incubating/manual/opennlp.html#tools.tokenizer.

Armstrong, T.G. et al., 2009. Improvements that don't add up: ad-hoc retrieval results since 1998. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, pp. 601–610.

Baeza-Yates, R. & Ribeiro-Neto, B., 1999. *Modern information retrieval*, ACM press New York.

Bauer, M., Konjugation (Grammatik). , 2013(02.09). Available at: http://www.uni-protokolle.de/Lexikon/Konjugation_(Grammatik).html.

Berger, A.L., Pietra, V.J. Della & Pietra, S.A. Della, 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1), pp.39–71. Available at: http://dl.acm.org/citation.cfm?id=234285.234289 [Accessed November 24, 2013].

Bick, E., 2004. A Named Entity Recognizer for Danish. In *LREC*.

Bird, S., 2004. Phonology. In R. Mitkov, ed. *The Oxford handbook of computational linguistics*. Oxford: Oxford University Press, pp. 3–24.

Bird, S. & Ellison, T.M., 1994. One-level phonology: Autosegmental representations and rules as finite automata. *Computational Linguistics*, 20(1), pp.55–90.

Bodenreider, O., 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1), pp.D267–D270.

Bouadjenek, M.R. et al., 2013. Evaluation of Personalized Social Ranking Functions of Information Retrieval. In *Web Engineering*. Springer, pp. 283–290.

Bubenhofer, N., 2011. Einführung in die Korpuslinguistik: Praktische Grundlagen und Werkzeuge. . , 2013(15.07). Available at: http://www.bubenhofer.com/korpuslinguistik/.

Buckley, C. et al., 2006. Bias and the limits of pooling. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp.619–620.

Bussmann, H., 2002. *Lexikon der Sprachwissenschaft. Dritte, aktualisierte und erweiterte Auflage*, Stuttgart: Kroener Verlag.

Carstensen, K.-U. et al., 2010. *Computerlinguistik und Sprachtechnologie*, Springer DE.

Castilho, F.M.B.M. et al., 2012. Corpus+ WordNet Thesaurus Generation for Ontology Enriching N. Calzolari, ed. *LREC - The International Conference on Language Resources and Evaluation*.

Charles, R.H., 2001. ACCOUNTING FOR USERS' INFLATED ASSESSMENTS OF ONLINE CATALOG SEARCH PERFORMANCE AND USEFULNESS: AN EXPERIMENTAL STUDY. *Information Research: an international electronic journal*, 6(2), p.101. Available at: http://www.doaj.org/doaj?func=openurl&issn=13681613&date=2001&volume=6&issue=2&spage=101&genre=article.

Chim, H. & Deng, X., 2007. A new suffix tree similarity measure for document clustering. *Proceedings of the 16th international conference on World Wide Web*, pp.121–130.

CIKM, 2013. ACM International Conference on Information and Knowledge Management. , 2013(17.07). Available at: http://www.cikm2013.org/index.php.

CiteULike, 2012. Scripting CiteULike. , 2013(27.07). Available at: http://wiki.citeulike.org/index.php/Importing_and_Exporting#Scripting_CiteULike.

Cleverdon, C.W., 1991. The significance of the Cranfield tests on index languages. *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pp.3–12.

Clinchant, S., 2012. Concavity in IR models. In *Proceedings of the 21st ACM international conference on Information and knowledge management (CIKM)*. ACM, pp. 2539–2542.

Computerlinguistik, I. für, 2013. Was ist Computerlinguistik U. Heidelberg, ed. , 2013(12.08). Available at: http://www.cl.uni-heidelberg.de/interest/ [Accessed December 8, 2013].

Croft, W.B., Metzler, D. & Strohman, T., 2010. *Search Engines - Information Retrieval in Practice* , Upper Saddle River, NJ, USA: Pearson, Inc.

Van Damme, C., Coenen, T. & Vandijck, E., 2001. Deriving a Lightweight Corporate Ontology form a Folksonomy: a Methodology and its Possible Applications. *Scalable Computing: Practice and Experience*, 9(4).

Demartini, G., 2007. Leveraging semantic technologies for enterprise search. *Proceedings of the ACM first PhD workshop in CIKM on PIKM 07*, p.25. Available at: http://portal.acm.org/citation.cfm?doid=1316874.1316879.

Dou, Z., Song, R. & Wen, J.-R., 2007. A large-scale evaluation and analysis of personalized search strategies. *Proceedings of the 16th international conference on World Wide Web*, pp.581–590.

Duden, 2013. Duden online. , 2013(19.09). Available at: www.duden.de.

Fellbaum, C., 2012. WordNet Search - 3.1. , 2013(22.09). Available at: http://wordnetweb.princeton.edu/perl/webwn.

Foundation, A.S. ed., 2012. BM25 Similarity Class in Apache Lucene 4.0. , 2013(31.07). Available at: http://lucene.apache.org/core/4_0_0/core/org/apache/lucene/search/similarities/BM25Similarity.html.

Garrido, G. et al., 2010. Information retrieval baselines for the ResPubliQA task. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments*. Springer, pp. 253–256.

Hamp, B. & Feldweg, H., 1997. Germanet-a lexical-semantic net for german. In *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*. Citeseer, pp. 9–15.

Hanks, P., 2004. Lexicography. In R. Mitkov, ed. *The Oxford handbook of computational linguistics*. Oxford: Oxford University Press, pp. 48–69.

Harpale, A. et al., 2010. CiteData: a new multi-faceted dataset for evaluating personalized search performance. *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp.549–558.

Huang, Y., Ma, X. & Li, D., 2010. Research and Application of Enterprise Search Based on Database Security Services. Available at: https://www.academypublisher.com/~academz3/proc/isnns10/papers/isnns10p238.pdf [Accessed November 23, 2013].

Hull, D.A., 1996. Stemming algorithms: A case study for detailed evaluation. *JASIS*, 47(1), pp.70–84.

Hutchins, W.J., 1986. *Machine translation: past, present, future*, Ellis Horwood Chichester.

Informationswissenschaft, 2013. Computerlinguistik - Die Informationswissenschaft in Begriffen U. des Saarlandes, ed. , 2013. Available at: http://server02.is.uni-sb.de/trex/index.php?id=3.1.

Inneren, B. des, 2010. Datenschutz in der Arbeitswelt - Eckpunktepapier zum Beschäftigtendatenschutz. Available at: http://www.bmi.bund.de/SharedDocs/Downloads/DE/Gesetzestexte/Entwuerfe/eckpunkte_an_datenschutz.html.

Jarmasz, M., 2012. Roget's thesaurus as a lexical resource for natural language processing. *arXiv preprint arXiv:1204.0140*.

Jarmasz, M. & Szpakowicz, S., 2001. The design and implementation of an electronic lexical knowledge base. In *Advances in Artificial Intelligence*. Springer, pp. 325–334.

Jaspers, A., 2012. Die EU-Datenschutz-Grundverordnung. *Datenschutz und Datensicherheit - DuD*, 36(8), pp.571–575. Available at: http://dx.doi.org/10.1007/s11623-012-0182-7.

Jin, R. & Si, L., 2004. A Bayesian approach toward active learning for collaborative filtering. *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pp.278–285.

Joel, W.R. et al., 2006. TF-ICF: A New Term Weighting Scheme for Clustering Dynamic Data Streams. In *Machine Learning and Applications, 2006. ICMLA '06. 5th International Conference on*. pp. 258–263.

Jurafsky, D. et al., 2000. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*, MIT Press.

Kamvar, S. et al., 2003. Extrapolation Methods for Accelerating PageRank Computations. *Twelfth International World Wide Web Conference (WWW 2003)*. Available at: http://ilpubs.stanford.edu:8090/865/.

Kilgarriff, A. & Yallop, C., 2000. What's in a Thesaurus? In *LREC*.

Kohn, A., Bry, F. & Manta, A., 2008. Exploiting a Company's Knowledge: The Adaptive Search Agent YASE. *I-KNOW-Knowledge Management and ...*. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.139.4658&rep=rep1&type=pdf [Accessed April 13, 2014].

Loth, R. et al., 2010. Linguistic information extraction for job ads (SIRE project). *Adaptivity, Personalization and Fusion of Heterogeneous Information*, pp.222–224.

Lovins, J.B., 1968. *Development of a stemming algorithm*, MIT Information Processing Group, Electronic Systems Laboratory.

Manning, C. et al., 2013. Stanford Tokenizer. Available at: http://nlp.stanford.edu/software/tokenizer.shtml.

Manning, C.D., Raghavan, P. & Schütze, H., 2008. *Introduction to information retrieval*, Cambridge University Press Cambridge.

Manning, C.D. & Schuetze, H., 1999. *Foundations of Statistical Natural Language Processing*, MIT Press.

Masterman, M., 1957. The thesaurus in syntax and semantics. *Mechanical Translation*, 4(1-2), pp.35–43.

Mikheev, A., 2004. Text Segmentation. In R. Mitkov, ed. *The Oxford handbook of computational linguistics,*. Oxford: Oxford Universit Press, pp. 201 – 218.

Millen, D.R., Feinberg, J. & Kerr, B., 2006. Dogear: Social bookmarking in the enterprise. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, pp. 111–120.

Miller, G.A. et al., 1990. Introduction to wordnet: An on-line lexical database*. *International journal of lexicography*, 3(4), pp.235–244.

Miller, G.A., 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11), pp.39–41.

Mitkov, R., 2004. *The Oxford handbook of computational linguistics*, Oxford: Oxford University Press Oxford.

Müller-Riedlhuber, H., 2009. The European Dictionary of Skills and Competences (DISCO): an example of usage scenarios for ontologies. *Proceedings of the 5th International Conference on Semantic Systems ISEMANTICS*, pp.467–479.

Müller-Riedlhuber, H. & Ziegler, P., 2012a. DISCO II - Prospects and challenges of a multilingual skills terminology. *DISCO II Final Conference*. Available at: http://disco.fwd.at/disco2_portal/images/Presentation_DISCO_II_Mueller_Riedlhuber_Ziegler_3s.pdf.

Müller-Riedlhuber, H. & Ziegler, P., 2012b. DISCO Thesaurus Explorer. , 2013(03.04). Available at: http://disco-tools.eu/disco2_portal/terms.php.

Müller-Riedlhuber, H. & Ziegler, P., 2012. DISCO-Projekt-Informationen. , 2013(03.04). Available at: http://disco-tools.eu/disco2_portal/projectInformation.php.

Nadeau, D. & Sekine, S., 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1), pp.3–26.

Navigli, R., 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2), p.10.

NIST, 2012. Full version of TREC 2012 test topics. , 2013(17.07). Available at: http://trec.nist.gov/data/web/12/full-topics.xml.

NIST, 2010. Text Retrieval Conference (TREC) Overview. , 2013(15.07). Available at: http://trec.nist.gov/overview.html.

Northwestern University Information Technology, 2013. Morph Adorner - English Lemmatization Process. Available at: http://picard.at.northwestern.edu/morphadorner/lemmatizer/lemmatizationprocess/.

Papacharissi, Z., 2009. The virtual geographies of social networks: a comparative analysis of Facebook, LinkedIn and ASmallWorld. *New Media & Society*, 11(1-2), pp.199–220. Available at: http://nms.sagepub.com/content/11/1-2/199.abstract.

Perera, P. & Witte, R., 2005. A self-learning context-aware lemmatizer for German. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05*. Morristown, NJ, USA: Association for Computational Linguistics, pp. 636–643. Available at: http://dl.acm.org/citation.cfm?id=1220575.1220655 [Accessed April 1, 2014].

Perez-Iglesias, J. et al., 2009. Integrating the probabilistic models BM25/BM25F into Lucene. *arXiv preprint arXiv:0911.5046*.

Porter, M.F., 1980. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3), pp.130–137. Available at: http://www.emeraldinsight.com/journals.htm?issn=0033-0337&volume=14&issue=3&articleid=1670983&show=html [Accessed March 30, 2014].

Porter, M.F., 2005. Porter stemmer in Java. Available at: http://tartarus.org/martin/PorterStemmer/java.txt.

Powers, D.M.W., 2011. Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1), pp.37–63.

Raghavan, V., Bollmann, P. & Jung, G.S., 1989. A critical investigation of recall and precision as measures of retrieval system performance. *ACM Transactions on Information Systems (TOIS)*, 7(3), pp.205–229.

Reichhold, M. et al., 2012. Automatic Generation of User Role Profiles for Optimizing Enterprise Search. *24th International Conference on SOFTWARE & SYSTEMS ENGINEERING and their APPLICATIONS*, 24, pp.241–248.

Reichhold, M., Kerschbaumer, J. & Fliedl, G., 2011. Optimizing enterprise search by automatically relating user context to textual document content. *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*, pp.1–6.

Robertson, S. & Zaragoza, H., 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Information Retrieval*, 3(4), pp.333–389.

Robertson, S.E., 1997. Overview of the okapi projects. *Journal of Documentation*, 53(1), pp.3–7.

Roget, P., 1852. Thesaurus of English Words and Phrases Longman. *Brown, Green, and Longmans New York*.

Roth, S., 2006. *Lexikalisch-semantische Netze: Anwendungsperspektiven für die Computerlinguistik*.

Sanderson, M., 2010. *Test collection based evaluation of information retrieval systems*, Now Publishers Inc.

SORNLERTLAMVANICH, V. et al., 2007. Statistical-Based Approach to Non-segmented Language Processing. *IEICE TRANSACTIONS on Information and Systems*, E90-D(10), pp.1565–1573. Available at: http://search.ieice.org/bin/summary.php?id=e90-d_10_1565&category=D&year=2007&lang=E&abst= [Accessed November 23, 2013].

Sproat, R., 2005. What is Computational Linguistics? , 2013(12.08). Available at: http://www.aclweb.org/archive/misc/what.html.

Taylor, P., 2009. *Text-to-speech synthesis*, Cambridge University Press.

Trost, H., 2004. Computational Morphology. In R. Mitkov, ed. *The Oxford handbook of computational linguistics*. Oxford: Oxford University Press, pp. 25–47.

Trotman, A. & Keeler, D., 2011. Ad hoc IR: not much room for improvement. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, pp. 1095–1096.

Uszkoreit, H., 1996. WHAT IS COMPUTATIONALLINGUISTICS? , 2013(12.08). Available at: http://www.coli.uni-saarland.de/~hansu/what_is_cl.html.

Vallet, D., Cantador, I. & Jose, J.M., 2010. Personalizing web search with folksonomy-based user and document profiles. In *Advances in Information Retrieval*. Springer, pp. 420–431.

Vater, H., 2002. *Einführung in die Sprachwissenschaft*, Wilhelm Fink Verlag. Available at: http://books.google.at/books?id=OxYWdHbKFUkC.

Voorhees, E., 1999. The TREC Conferences: An Introduction. , 2013(22.07). Available at: http://trec.nist.gov/presentations/TREC8/intro/sld001.htm.

Voorhees, E.M., 2005. TREC: Improving Information Access through Evaluation. *Bulletin of the American Society for Information Science and Technology*, Vol. 32(No. 1). Available at: http://www.asis.org/Bulletin/Oct-05/voorhees.html.

Vossen, P., 2004. Eurowordnet: a multilingual database of autonomous and language-specific wordnets connected via an Inter-LingualIndex. *International journal of lexicography*, 17(2), pp.161–173.

White, R.W., Bennett, P.N. & Dumais, S.T., 2010. Predicting short-term interests using activity-based search context. *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp.1009–1018.

Willett, P., 2006. The Porter stemming algorithm: then and now. *Program: electronic library and information systems*, 40(3), pp.219–223. Available at: http://www.emeraldinsight.com/journals.htm?issn=0033-0337&volume=40&issue=3&articleid=1563486&show=html [Accessed March 30, 2014].

Xu, S. et al., 2008. Exploring folksonomy for personalized search. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 155–162.