Diplomarbeit

Informationsextraktion durch Zusammenfassung maschinell selektierter Textsegmente

Timm Euler

Diplomarbeit
am Fachbereich Informatik
der Universität Dortmund

Dortmund, 8. Oktober 2001

Betreuer:

Prof. Dr. Katharina Morik
Dipl.-Inform. Ralf Klinkenberg

Denn zwischen dem, ein Ding verstehen und ein Ding nicht verstehen, gibt es viele Klassen, in denen sich 9/10 *des menschlichen Geschlechts ganz commode aufhalten.*

—Georg Christoph Lichtenberg

Danksagung

Für die ausgezeichnete Betreuung dieser Diplomarbeit möchte ich mich bedanken bei Prof. Dr. Katharina Morik, die die Arbeit in eine erfolgreiche Richtung lenkte, und Dipl.-Inform. Ralf Klinkenberg, der nie um gute Ratschläge verlegen war.

Der Einsatz zweier Software-Pakete zur Sprachverarbeitung wurde nur ermöglicht durch Dr. Günter Neumann und Markus Becker vom Deutschen Forschungsinstitut für Künstliche Intelligenz in Saarbrücken (für Meson) und Andreas Mertens (für Wap), für deren Unterstützung bei technischen Fragen ich sehr dankbar bin. Stefan Rüping, dessen mySVM-Implementation von Support Vector Machines ich verwendete, half mir theoretisch wie praktisch beim Umgang mit SVMs.

Weiterhin bedanke ich mich bei den vielen Freunden und Bekannten, die mir ihre Emails zur Verfügung stellten und die Fragebögen beantworteten, teilweise mit erheblichem Zeitaufwand.

Nicht zuletzt danke ich meinen Eltern für die Ermöglichung eines spannenden Informatikstudiums.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1. Einleitung

Die Zahl der elektronisch verfügbaren Texte wächst seit der Erfindung des Computers kontinuierlich und ist spätestens mit der Verbreitung des World Wide Web unübersehbar geworden. Zur Suche nach Informationen in Texten dienen oft wiederum Computer, die dazu verschiedene Methoden der Klassifizierung, Filterung und Zusammenfassung verwenden. Auch menschliche Verwaltung von Textsammlungen, zum Beispiel bei Webverzeichnissen, spielt eine große Rolle. Die Weiterentwicklung der automatisierten Verfahren ist jedoch unerlässlich, soll nicht ein Großteil der eigentlich nutzbaren Texte unbeachtet bleiben.

Durch die Vernetzung von Computern kommt der Aspekt der Kommunikation hinzu, die aufgrund begrenzter Netzkapazitäten zum größten Teil textbasiert ist (Email) und seit Ende des 20. Jahrhunderts zur Massenkommunikation angewachsen ist. Dabei liegt heute ein heterogenes Angebot von Kommunikationswegen vor, die eine Person nutzen kann: Festnetztelefon, Mobiltelefon, Telefax, gesprochene Nachrichten, Email, SMS-Nachrichten für Mobiltelefone (SMS: Short Message Service), Online-Unterhaltungen (Chat) und Newsgroups. Die Vermittlung zwischen verschiedenen Modalitäten der Kommunikation gewinnt damit immer mehr an Bedeutung: Nachrichten sollen unabhängig von ihrem ursprünglichen Format den Benutzern in der Form präsentiert werden, die dem momentanen Lesegerät entspricht. Diese Zielvorstellung wird häufig mit dem Begriff *Unified Messaging* bezeichnet ([Arbanowski und van der Meer 1999]). Solchen Systemen liegt oft das Agenten-Paradigma zugrunde, wobei Software-Agenten die Umwandlung einer Nachricht in ein anderes Format übernehmen ([Abu-Hakima et al. 1996]).

Die Methoden dieser Diplomarbeit werden durch das Ziel der Umwandlung von Emailtexten in SMS-Nachrichten motiviert, wobei für SMS eine protokollbedingte strikte Maximallänge von 160 Zeichen des Latin-1-Standards gilt, so dass der ursprüngliche Text in den meisten Fällen gekürzt werden muss. Dabei sollen nur Emails verwendet werden, die für den Anwender von einer gewissen Dringlichkeit sind, und innerhalb dieser Emails muss wiederum zwischen wichtigen und weniger wichtigen Informationen unterschieden werden, um dem stark begrenzten Platz in einer SMS-Nachricht Rechnung zu tragen. Es werden also *inhaltliche* Kriterien zur Selektion interessierender Textsegmente benötigt. Damit bewegt sich diese Diplomarbeit auf dem Gebiet der Informationssuche in Texten.

Mit der sinnvollen Verkürzung von Texten beschäftigt sich auch der Bereich der automatischen Zusammenfassung von Texten. Heutige Verfahren dafür erstellen meistens Extrakte, also ausgewählte Teile des Originals, als Zusammenfassung. Die meisten Arbeiten auf diesem Gebiet befassen sich mit *generischen* Extrakten, die den Text als Ganzes wiedergeben sollen. Es werden also Textteile gesucht, die möglichst informativ und repräsentativ für den ganzen Text sind. Dies wird in Kapitel 3 beschrieben. Im Gegensatz dazu steht hier die fokussierte Suche nach wichtigen (dringenden) Informationen zu einem vorgegebenen Thema im Vordergrund. Mit der Themenvorgabe wird die Selektion *gezielt* und ähnelt eher der *Informationsextraktion*, die nur nach einzelnen Fakten in Texten sucht (Abschnitt 3.4). Zur Informationssuche in Texten stellt diese Diplomarbeit also die *gezielte Extraktion* von Textsegmenten (Sätzen) vor (Kapitel 4). Dazu werden Wortlisten benutzt, die automatisch aus markierten Beispieltexten bestimmt werden, wobei die Wörter das vorgegebene Thema repräsentieren und zum Auffinden der interessierenden Sätze dienen.

Ein Vorteil der gezielten gegenüber der generischen Extraktion ist die bessere Bewertbarkeit anhand der vorher bekannten Aufgabenstellung. Generische Extrakte sollen im wesentlichen für alle Zwecke nutzbar sein, die auch der Ursprungstext erfüllt hätte, was schwierig zu bemessen ist. Bei gezielter Extraktion vereinfacht sich die Bewertung, da bekannt ist, welche Informationen im Extrakt enthalten sein sollten. Der Informationsgehalt kann also mit gezielten Fragen an Leser der Extrakte untersucht werden. Dies wird ebenfalls in dieser Arbeit überprüft (Kapitel 6).

Wie sich herausstellt, müssen jedoch die extrahierten Textteile in vielen Fällen noch weiter komprimiert werden, um die sehr platzbeschränkten SMS-Nachrichten erstellen zu können. Daher kommt in dieser Diplomarbeit eine zweite Stufe der Zusammenfassung hinzu. Dazu werden aus den ausgewählten Sätzen unwichtige Teile entfernt (Kapitel 5). Das Verfahren nähert sich damit noch weiter der Informationsextraktion an, was den Titel der Arbeit erklärt.

Zusammengefasst, werden in dieser Arbeit zwei Stufen entwickelt, die die Kürzung von Texten unter einem inhaltlichen Gesichtspunkt ermöglichen. Die erste Stufe ist die Auswahl von einzelnen Sätzen aus beliebigen Texten unter dem Gesichtspunkt ihrer Zugehörigkeit zum vorgegebenen Thema. Die Untersuchung der dazu entwickelten Verfahren aus Kapitel 4 steht im Mittelpunkt der Arbeit. Die zweite Stufe kann zusätzlich zur ersten eingesetzt werden, wenn die Anwendung es erfordert; hier werden die extrahierten Sätze gekürzt, möglichst ohne dass wichtige Informationen verloren gehen. Für die Umwandlung von Emails in SMS-Nachrichten ist diese Stufe erforderlich; allgemein kann jedoch die gezielte Auswahl von informativen Sätzen bereits genügen.

Die Hauptfragen, die in dieser Arbeit untersucht werden, können daher wie folgt formuliert werden:

- Lassen sich mit den vorzustellenden Methoden Wortlisten bestimmen, deren Wörter für das vorgegebene Thema charakteristisch sind?
- Ist das Auffinden der Sätze, die die gesuchten Informationen enthalten, mit den Wortlisten möglich? Wieviel Information wird gefunden, wieviel verpasst?
- Wie kann innerhalb von Sätzen zwischen wichtiger Information und weniger interessantem Beiwerk unterschieden werden, und wie stark lassen sich Sätze damit komprimieren, ohne unverständlich zu werden?
- Ist wegen der Einschränkung bei der Informationssuche auf ein bestimmtes Thema eine fundiertere Untersuchung des Informationsgehaltes der entstehenden Kurztexte möglich?

Der folgende Abschnitt gibt einen Überblick über den Aufbau dieser Arbeit, aus dem hervorgeht, auf welche Weise die Arbeit sich diesen Fragen nähert.

1.1. Aufbau der Arbeit

Da zur Informationssuche eine gewisse sprachliche Verarbeitung der Texte, in denen gesucht wird, notwendig ist, führt Kapitel 2 kurz in das Gebiet der Verarbeitung natürlicher Sprache mit dem Computer ein. Die Beschreibung dient nur dem Verständnis der für diese Arbeit wichtigen Aspekte und ist daher recht oberflächlich; für Interessierte wird auf die angegebene Literatur verwiesen. Im Rahmen dieser Arbeit wird die sprachliche Verarbeitung von zwei fertigen Werkzeugen übernommen, Meson und Wap, deren Funktionsweisen in diesem Kapitel erläutert werden.

Kapitel 3 behandelt ausführlich verschiedene Methoden zur Suche nach Informationen in Texten. Insbesondere wird das Verhältnis dieser Arbeit zu den Gebieten der Informationsextraktion und der automatischen Textzusammenfassung beleuchtet. Verschiedene Ansätze aus der Textzusammenfassung werden überprüft auf ihre Tauglichkeit für die gezielte, also an einem Thema orientierte Suche nach Textsegmenten. Es wird deutlich, dass eigene Methoden dazu entwickelt werden müssen. Dies geschieht in Kapitel 4, das die automatische Erstellung von Listen von Wörtern zu einem Thema behandelt; anschließend wird erläutert, wie mit diesen Listen die Auswahl von Sätzen erfolgen kann. Dieses Kapitel behandelt also die erste der oben angesprochenen Stufen.

Im folgenden Kapitel 5 wird die zweite Stufe entwickelt. Zur weiteren Komprimierung von Sätzen werden unwichtige Teile, die also nicht das vorgegebene Thema behandeln, entfernt. Außerdem stellt dieses Kapitel das Verfahren vor, mit dem die gekürzten Sätze zu einer SMS-Nachricht zusammengesetzt werden. Schließlich werden Überlegungen zum alleinigen Einsatz der zweiten Stufe, ohne die erste, angestellt.

Kapitel 6 beschreibt die Experimente und Resultate zur Auswertung der Verfahren aus den Kapiteln 4 und 5. Die Auswertung geschieht auf zwei Textsammlungen, die zunächst vorgestellt werden. Für eine davon ist nur die erste Stufe der Satzauswahl interessant. Der Schwerpunkt der Experimente liegt auf der anderen Sammlung, die aus Emails besteht. Diese Emails werden mit den Verfahren beider Stufen zu SMS-Nachrichten umgewandelt. Der Erfolg des Vorgehens wird auch anhand des Informationsgehaltes der SMS gemessen.

Das letzte Kapitel 7 fasst die Resultate zusammen und gibt Anregungen für weitere Einsatzmöglichkeiten und Varianten der Verfahren dieser Arbeit. Im Anhang finden sich Auszüge aus einigen der automatisch errechneten Themenwortlisten sowie einige kurze Hinweise zur Implementation für interessierte Anwender.

2. Verarbeitung natürlicher Sprache

Dieses Kapitel bietet eine kurze Einführung in die Verarbeitung natürlicher Sprache mit dem Computer. Die gängige Bezeichnung für dieses Gebiet ist NLP für *Natural Language Processing*. Abschnitt 2.1 erläutert die wichtigsten Komponenten eines sprachverarbeitenden Systems. In Abschnitt 2.2 geht es um flache Technologien und das Parsen mit Wortagenten, da diese den beiden im Rahmen dieser Arbeit eingesetzten NLP-Werkzeugen, MESON (Nachfolger von SMES, [NEUMANN et al. 1997]) und WAP ([MERTENS 1997]), zugrundeliegen.

2.1. NLP

Natürliche Sprachen sind von Menschen verwendete Sprachen im Gegensatz zu formalen, künstlichen Sprachen. Sie sind Gegenstand der Linguistik (Sprachwissenschaft). Sie werden mündlich und schriftlich verwendet. Gesprochene Sprache wird mittels *Spracherkennung* in Text oder geeignete Kodierungen überführt. Die eigentlichen NLP-Techniken setzen diese Stufe voraus bzw. behandeln geschriebenen Text, der auch schon in elektronischer Form vorliegt (gedruckte Texte automatisch in elektronische umzuwandeln, ist Aufgabe der Schrifterkennung). Kennzeichnend für NLP-Systeme ist es, Wissen über Sprache zu verwenden, anders als beispielsweise bei rein statistischen Ansätzen, die damit nicht unter NLP fallen.

In der Linguistik werden Sprachen auf mehreren Ebenen beschrieben, die allerdings nicht unabhängig voneinander sind. Die Phonologie befasst sich mit den Lauten einer Sprache und ist daher hauptsächlich für die Spracherkennung wichtig, wenngleich der Klang eines Satzes auch auf anderen Ebenen eine Rolle spielt. Die für NLP hauptsächlich relevanten Beschreibungsebenen sind Morphologie, Syntax und Semantik. Vereinfacht gesagt, befasst sich die Morphologie mit der Struktur von Wörtern, während die Syntax die Struktur von Sätzen analysiert. In der Semantik geht es um die Bedeutung von sprachlichen Äußerungen—deren Abhängigkeit von der Situation, in der diese gemacht werden, schließlich in der Pragmatik untersucht wird. Die Kenntnis dieser Ebenen hilft, die folgenden typischen Komponenten eines sprachverarbeitenden Systems einzuordnen, von denen die meisten auch in MESON und WAP enthalten sind.

- Die Vorverarbeitung eines Textes abstrahiert von seiner elektronischen Kodierung sowie eventuell vorhandenen Formatierung. Ein *Tokenisierer* zerlegt den Eingabetext in einzelne *Token*, das sind Zeichenketten, die als Kandidat für ein Wort oder Satzzeichen gelten. Oft werden auch die Satzgrenzen schon in diesem Modul bestimmt; dies kann jedoch auch dem Parser vorbehalten bleiben.

- Im Lexikon sind die Wörter der Sprache enthalten, meist mit Zusatzinformationen, die verschiedene Formen desselben Wortes angeben (`lief` als eine Form von `laufen`) oder die Zusammenarbeit mit anderen Teilen des NLP-Systems ermöglichen. Ebenso sind für jedes Wort die möglichen Wortarten angegeben (Nomen, Verb, Adjektiv usw.). Das Lexikon liegt auf der Ebene der Morphologie.

- Der Tagger versucht, aus den möglichen Wortarten eines Wortes die im aktuellen Zusammenhang richtige auszuwählen. Zum Beispiel ist `Halt` in `Nächster Halt: Dortmund` ein Nomen, aber in `Halt das mal eben!` ein Verb. Diese Verfahren liegen zwischen Morphologie und Syntax.

- Der Parser versucht, einer Folge von Wörtern eine Struktur zuzuordnen, die einer vorgegebenen Grammatik entspricht. Damit gehört das Parsen zur Syntax.

Ob semantische Komponenten eingebaut sind, und welche, hängt stark vom System und seinen vorgesehenen Einsatzmöglichkeiten ab. MESON enthält keine semantische Verarbeitung, während WAP eine Komponente zum Aufbau semantischer Netze hat, die aber im Rahmen dieser Arbeit nicht verwendet wurde.

Zu jeder dieser NLP-Komponenten existieren sehr viele Forschungsarbeiten und praktische Implementierungen. Hier sollen nur einige generelle Aspekte, die bei MESON und WAP eine Rolle spielen, kurz angesprochen werden.

Vorverarbeitung

Die Zerlegung in Token wie auch die Bestimmung von Satzgrenzen beruhen auf regulären Ausdrücken. Die Aufgabe ist nicht trivial, zum Beispiel sind in Sprachen wie Chinesisch die Wortgrenzen nicht direkt aus der Schreibweise erkennbar. In den europäischen Sprachen kann ein Punkt zu einer Ordnungszahl, einer Datumsangabe, einer Internetadresse oder einer Abkürzung gehören, ohne ein Satzende zu bezeichnen, er kann aber auch gleichzeitig ein Satzende anzeigen. Trennstriche sind von Gedankenstrichen, die sich zufällig am Zeilenende befinden, zu unterscheiden. Ein geringer Prozentsatz an Fehlern muss bei diesen Problemen in Kauf genommen werden (siehe [GREFENSTETTE und TAPANAINEN 1994]). Bei MESON und WAP werden Internetadressen und Trennstriche nicht erkannt. MESON besitzt eine gute Erkennung von Zeit- und Datumsangaben und ersetzt einige Abkürzungen automatisch durch ihre längere Version, wobei der Punkt immer entfernt wird, so dass einige Satzenden, die mit Abkürzungen zusammenfallen, nicht erkannt werden.

Lexikon

Die Aufgabe des Lexikons ist es, ein Token zu einem gespeicherten Wort mit all seinen Zusatzinformationen zuzuordnen, und dies effizient. Ist das Token nicht im Lexikon, so muss das NLP-System mit der fehlenden Information umgehen können, denn kein Lexikon enthält alle Wörter und Wortformen, die in einem freien Text vorkommen können (man denke an Tipp- und Rechtschreibfehler, Eigennamen und Neubildungen).

Unterschiede zwischen Lexika entstehen durch die Art der zu einem Wort gespeicherten Information. Diese steht in engem Zusammenhang mit der Art des Parsens. Oft genügt es, die möglichen Wortarten eines Wortes abzuspeichern; andere Parser benötigen sehr viel mehr Information für jedes Wort, wie bei WAP (siehe unten).

Auch die Organisation des Lexikons kann unterschiedlich ausfallen: Vollformenlexika speichern jede mögliche Wortform ab (`gehe, gehst, geht, ...`), doch kann man auch die Regelmäßigkeiten der Sprache (hier auf Ebene der Morphologie) ausnutzen und Wortstrukturen analysieren, um zum Beispiel bei `gehen` die Suffixe `-e, -st, -t` abzutrennen, so dass nur eine *Stammform* (abgesehen von unregelmäßigen Formen) im Lexikon gespeichert werden muss. Ein bekanntes und effizientes Verfahren dafür beruht auf endlichen Automaten ([KOSKENNIEMI 1984]). Es wird in MESON im Rahmen der morphologischen Verarbeitung durch das Modul MORPHIX eingesetzt (vgl. [NEUMANN et al. 1997]). Eine Stärke von MORPHIX ist die Analyse von Komposita (zusammengesetzten Nomen), für die kein eigener Eintrag im Lexikon notwendig ist. Stattdessen werden sie automatisch aufgeteilt in längstmögliche Teile, die im Lexikon enthalten sind. Für Sprachen wie Deutsch, die reich an Flektionen sind und gerne Komposita bilden, ist eine gute morphologische Komponente sehr sinnvoll. Eine andere Methode nutzt WAP: Die Wörter sind in Klassen eingeteilt, die gemeinsame morphologische Merkmale haben, so dass zu jedem Wort nur seine Klasse abgespeichert ist.

Mit Hilfe des Lexikons lassen sich also verschiedene Formen eines Wortes auf dieselbe Form zurückführen. Dies nennt man *Stammformenreduktion* (vgl. Abschnitt 3.2). Manchmal wird dies auch ohne Lexikon durchgeführt, durch einfaches Abschneiden von Wortenden, die aussehen wie Flektionen. Beispielsweise könnte man die Endung `-st` entfernen, wie sie deutsche Verben in der zweiten Person Singular aufweisen. Dies nennt man *Stemming*; beliebte Verfahren dafür (Stemmer) werden in [LOVINS 1968] und [PORTER 1980] beschrieben. Stemming ist aber ein sehr grobes Verfahren, bei dem Wortart und Wortbedeutung nicht berücksichtigt werden (vergleiche das Wort `Rast`). Schon bei der flektionsarmen englischen Sprache tauchen Probleme auf, wie die Verkürzung von `news` zu `new` deutlich macht; für flektionsreiche Sprachen wie Deutsch empfehlen sich Stemmer nicht.

Tagger

Die Wortart eines Wortes bestimmt sich aus seiner Funktion im Satz und ist damit kontextabhängig. Die richtige Wortart wird vom *Part of speech-Tagger* oder kurz Tagger bestimmt. Hierfür existieren rein statistische wie auch linguistisch motivierte sowie hybride Verfahren. Eine Übersicht findet sich in [ABNEY 1997]. Allerdings verwenden weder MESON noch WAP eine eigene Komponente zum Taggen. In kommenden Versionen von MESON wird ein Tagger nach [BRILL 1995] eingebaut sein. Bei WAP wird die Wortart auf ganz andere Art bestimmt, siehe unten.

Parser

Für das Parsen natürlicher Sprache existieren höchst unterschiedliche Verfahren, die auf unterschiedlichen grammatischen Formalismen beruhen und damit teilweise auch verschiedene Sichtweisen der Funktionsweise menschlicher Sprache umsetzen. Ich werde hier nur den Zusammenhang erläutern, der die Notwendigkeit der flachen Technologien deutlich macht, die im folgenden Abschnitt (2.2) beschrieben werden.

Die Frage, ob ein sprachlicher Satz der gegebenen Grammatik genügt, entspricht dem Wortproblem formaler Sprachen (für eine Einführung in formale Sprachen und Komplexitätstheorie siehe [Wegener 1993]). Das Wortproblem für reguläre und kontextfreie Sprachen ist in polynomieller Zeit in der Länge des Wortes und der Größe der Grammatik lösbar, während es für kontextsensitive Sprachen NP-vollständig ist. Die meisten, vielleicht auch alle Strukturen menschlicher Sprachen sind höchstens kontextfrei[1]; dementsprechend gibt es Parser, die auf solchen Formalismen beruhen. Ein bekanntes Beispiel ist der Earley-Algorithmus ([Earley 1987]).

Reguläre Grammatiken reichen nicht aus, um natürliche Sprachen ganz abzudecken: Die rekursive Verschachtelung von Teilsätzen ineinander ist ein häufiges Phänomen in menschlichen Sprachen, das mit regulären Grammatiken nicht modellierbar ist.

Wie effizient ein kontextfreier Parser aber in der Praxis ist, hängt von der Komplexität der gegebenen Grammatik ab. Natürliche Sprachen bieten genügend Regeln, Ausnahmen von Regeln und Ausnahmen von Ausnahmen, um eine Grammatik mit Anspruch auf Vollständigkeit sehr groß werden zu lassen. Mit steigender Ausdrucksstärke der Grammatik wird aber das Parsen immer ineffizienter, da die Anzahl der möglichen Analysen eines Satzes mit seiner Länge exponentiell steigt. Bei langen Sätzen, wie sie zum Beispiel in Zeitungsartikeln häufig sind, oder bei großen Textsammlungen ist die Laufzeit des Parsers daher nicht mehr akzeptabel.

Ein weiteres Problem ist die Robustheit des Parsens. Wenn ein Satz nicht vollständig geparst werden kann, was bei längeren Texten häufig vorkommt, sollten zumindest Teilstrukturen erkannt werden. Das Parsen eines Satzes sollte nicht wegen eines Tippfehlers ganz scheitern müssen. Fehlertoleranz ist also ein wichtiges Kriterium, ist aber bei kontextfreien Parsern mit relativ viel technischem Aufwand verbunden.

2.2. Flache Techniken und Parsen mit Wortagenten

Mit der weiten Verbreitung von Computern und später dem Wachstum des Internet hat sich die Anzahl der elektronisch verfügbaren Texte sehr stark vergrößert. Damit einher ging die Notwendigkeit, große Textsammlungen (Korpora) sinnvoll zu verwalten und die Suche nach Informationen mit Computern zu unterstützen. Sprachverarbeitende Systeme stehen also vor der Aufgabe, schnell und effizient viele Texte zu verarbeiten und ihnen verwendbare Strukturen zuzuordnen. Das hat zu einem Wechsel der hauptsächlichen Ausrichtung der NLP-Forschung geführt (vgl. [Jacobs und Rau 1993]): Forschte man früher an möglichst guten Grammatiken in verschiedenen Formalismen, an guten semantischen Modellen und trickreichen Parsern, so waren nun aus Effizienzgründen schwächere, aber schnellere Technologien notwendig, die dennoch sinnvolle Ergebnisse liefern mussten. Jacobs und Rau zogen 1993 das Fazit:

> „As in AI in general, scale-up is the forcing function behind the shift in natural language research. The demands of real text applications are not compatible with the relatively deep, computation-intensive, sentence-driven methods of analysis [of earlier methods].“[2]

[1] Einige sprachliche Konstruktionen, die streng genommen kontextsensitiv sind, existieren beispielsweise im Schweizerdeutsch ([Shieber 1987]) oder in der afrikanischen Sprache Bambara ([Culy 1985]).

[2] vgl. [Jacobs und Rau 1993], S. 145

Mit „sentence-driven“ ist gemeint, dass frühere Methoden oft nur an einzelnen Sätzen entwickelt und getestet wurden, nicht an beliebig langen Texten (ibd.). Die Bezeichnung *Shallow Text Processing* steht für Techniken, die in heutigen, gut skalierenden Systemen eingesetzt werden; dabei ist die wesentlich veränderte Komponente der Parser.

Flache Parser (auch *Chunk Parser* oder *Partial Parser* genannt; eine Übersicht bietet [Abney 1997]) verwenden endliche Automaten, also reguläre Grammatiken, um Teilstrukturen in Sätzen zu erkennen. Zum Beispiel haben typische Nominalphrasen einen einfachen Aufbau (Artikel, optionale Adjektive, Nomen), der leicht mit endlichen Automaten modellierbar ist. Größere Strukturen lassen sich durch *Cascading* finden, also die Anwendung von mehreren endlichen Automaten nacheinander, die jeweils als Input den Output des vorigen Automaten[3] erhalten (siehe [Abney 1996]). Auch Satzstrukturen, also Haupt- und Nebensätze samt ihrer Abhängigkeiten, lassen sich so finden ([Neumann et al. 2000]). Die Laufzeit ist linear in der Anzahl der Wörter. Strukturen, die zu keinem vorgegebenen Pattern passen, bleiben unanalysiert, was die Robustheit der Verfahren sicherstellt.

Das Ergebnis nach dem Parsen ist also bei flacher Sprachverarbeitung eine mehr oder weniger lose Sammlung von Teilstrukturen statt eines kompletten Syntaxbaumes. Die Teilstrukturen sind jedoch für viele Anwendungen ausreichend, wie im folgenden Kapitel deutlich werden wird. Der Entwurf von Patterns bzw. Grammatiken für flaches Parsen ist zudem einfacher und kann auch leichter maschinell unterstützt werden (maschinelles Lernen wird dazu zum Beispiel in [Munoz et al. 1999] oder [Ramshaw und Marcus 1995] eingesetzt).

Meson verwendet kaskadierte endliche Automaten zum Finden von Nominal-, Präpositional- und Verbalphrasen. Eine Komponente zum Auffinden der Satzstruktur (Satzparser) soll in einer kommenden Version eingebaut werden.

WAP

Ein ganz anderer Ansatz beim Parsen liegt Wap zugrunde. Dieses System gehört zum Bereich des Wortagenten-Parsing ([Helbig und Mertens 1994]). Es beruht auf einem Lexikon, in dem sehr viele Informationen nicht nur über Wörter, sondern auch über deren genaue Verwendung abgelegt sind. Beispielsweise ist das Verb `sehen` transitiv, es erwartet also im Satz ein Akkusativobjekt, welches angibt, was gesehen wird (ein Subjekt, die Sehende, muss immer vorhanden sein). Eine Präposition wie `mit` erwartet eine nachfolgende Nominalphrase. Diese sogenannte Subkategorisierung kann im Lexikon abgelegt werden. Auch semantische Subkategorisierungen sind möglich: so erwartet das Verb `essen` als Subjekt einen Menschen, `fressen` dagegen ein Tier.

Die Idee von Wap als Wortagenten-Parser ([Mertens 1997]) besteht nun darin, von links nach rechts durch einen Satz zu gehen, bei jedem Wort dessen Erwartungen an vorige und nachfolgende Konstruktionen aufzusammeln und sie mit dem bisher Gefundenen abzugleichen. Jedes Wort wird als ein eigener Agent betrachtet, der Erwartungen an seine Verwendung hat. Viele Erwartungen können erst nach Einbeziehung späterer Wörter erfüllt werden. Zum Beispiel löst das Wort `der` als Artikel die Erwartung aus, dass mit diesem Artikel eine Nominalphrase eröffnet wird. Findet sich später ein Nomen, so ist

[3] Hier handelt es sich genau genommen um *Transducer*, also Automaten mit Ausgabe.

die Nominalphrase komplettiert. Beim Abgleich mit bisherigen Erwartungen werden die im Lexikon gespeicherten Subkategorisierungsinformationen verwendet.

Ein Wortagent kann auch mehrere Erwartungen auslösen, und in einem längeren Satz kann sich die Zahl der möglichen Lesarten, die von verschiedenen Wortagenten vorgeschlagen wurden und ihrer Komplettierung harren, stark erhöhen. Deshalb greifen immer dann, wenn diese Zahl zu groß wird, Heuristiken, die die meisten Lösungen wieder streichen. Die Auswahl wird so gesteuert, dass bereits komplettierte Phrasen behalten werden. Zusätzlich werden längere Phrasen bevorzugt. Eine genauere Beschreibung der Heuristiken findet sich in [Mertens 1997].

Dieses Verfahren ist nicht sehr schnell, da die heuristischen Algorithmen einige Zeit benötigen, diejenigen bisher erkannten Teilphrasen auszuwählen, die am vielversprechendsten sind. Es ist aber aus zwei Gründen ein sehr robustes Verfahren: Erstens, weil kurze Teilstrukturen auch als zulässiges Ergebnis zählen, so daß einzelne Wörter, die in keine Erwartung passen, keinen Fehler auslösen. Zweitens, weil unbekannte Wörter trotz der Abhängigkeit des Verfahrens vom Lexikon ebenfalls keinen Fehler auslösen, sondern *unterspezifizierte Hypothesen* erzeugen. Zum Beispiel löst in der Phrase `das Buch` das erste Wort die Erwartung aus, auf ein Nomen mit neutralem Geschlecht zu treffen, was von `Buch` erfüllt wird. Stünde dort aber wegen eines Tippfehlers `das Boch`, so würde `Boch` unterspezifizierte Hypothesen erzeugen, nach denen es ein Verb, Adjektiv oder Nomen sein könnte, deren Genus, Numerus usw. nicht festgelegt sind. Von diesen Hypothesen würde nur die Nomeninterpretation die Erwartung des ersten Wortes (Nominalphrase) erfüllen und deshalb ausgewählt werden.

Ein großer Vorteil von Wap gegenüber Meson ist, dass es nicht nur einzelne Phrasen findet und nebeneinander stellt, sondern sie auch zusammenfügt, wenn das möglich ist. Dank der Subkategorisierungsinformation können zum Beispiel den Verben ihre Argumente, also Subjekt, Akkusativobjekt und Dativobjekt, zugeordnet werden. So können einfache Sätze sogar komplett „tief" geparst werden, was bei Meson nicht möglich ist. Das ist der Grund, warum Wap zusätzlich eingesetzt wurde, um die Fähigkeiten von Meson zu ergänzen, wie in Kapitel 5 beschrieben wird.

3. Informationssuche in Texten

Im vorigen Kapitel (Abschnitt 2.2) wurde schon erwähnt, dass die zunehmende Notwendigkeit der Verarbeitung großer Textmengen zu neuen Aufgaben und Technologien geführt hat. Systeme, die Textanalyse nutzen, um Menschen die Auswertung solcher Textsammlungen zu erleichtern, werden heute unter dem Begriff *Text Mining* zusammengefasst, in Anlehnung an *Data Mining*, wo es um die Nutzung großer, aber strukturierter Datensammlungen geht. In diesem Kapitel werden die wesentlichen Aufgaben und Methoden solcher Systeme beschrieben. Nach einer Übersicht (Abschnitt 3.1) und der Behandlung von Textrepräsentationen (3.2) folgt ein Abschnitt über Textklassifikation (3.3), da einige solche Methoden in dieser Arbeit eingesetzt werden. Abschnitte über Informationsextraktion (3.4) und automatische Textzusammenfassung (3.5), die beiden Bereiche, denen diese Arbeit zugeordnet ist, schließen sich an.

3.1. Übersicht

Text Mining-Systeme werden nach ihrer Aufgabenstellung üblicherweise in die folgenden, sich teilweise überschneidenden Bereiche eingeteilt.

Information Retrieval (IR) Dies ist die Aufgabe, aus einer gegebenen Dokumentsammlung diejenigen Dokumente auszuwählen, die am besten zu einer Benutzeranfrage passen. Die Benutzeranfrage wird typischerweise in natürlicher Sprache formuliert, das System vergleicht die Anfrage mit den Dokumenten und wählt anhand eines Ähnlichkeitsmaßes Dokumente aus, die zum Thema der Anfrage passen. Den Benutzern bleibt die Suche durch viele irrelevante Dokumente erspart. Eine umfangreiche Einführung in dieses Gebiet ist [Baeza-Yates und Ribeiro-Neto 1999].

Information Filtering (IF) Hierbei ist die Dokumentsammlung nicht statisch, sondern verändert sich, wie dies etwa bei der Online-Ausgabe einer Tageszeitung der Fall ist, die täglich neue Artikel zum Archiv hinzufügt. Dagegen ist das Benutzerinteresse gleichbleibend auf bestimmte Themen gerichtet[1] und kann ausführlicher repräsentiert werden als bei spontanen Anfragen. Hier bietet [Oard 1997] eine gute Einführung.

Informationsextraktion Auf diesem Gebiet geht es um das Auffinden von spezifischen Informationen in einzelnen Texten, wobei andere Inhalte der Texte ignoriert werden können. Meistens ist es das Ziel, von bestimmten Sachverhalten, die in den Texten angesprochen werden können, eine maschinell verwendbare Repräsentation zu erhalten. Dies wird in Abschnitt 3.4 genauer erläutert.

[1] Auch zeitlich wechselnde Benutzerpräferenzen sind allerdings von Interesse, vgl. [Klinkenberg 1998].

Textzusammenfassung Die Zusammenfassung eines Textes soll Benutzern ermöglichen, eine rasche Übersicht über seinen Inhalt zu erhalten, um schnell entscheiden zu können, ob der ganze Text für sie interessant ist. Verfahren zur automatischen Zusammenfassung werden in Abschnitt 3.5 behandelt.

Textkategorisierung Hier soll eine Textmenge in Kategorien eingeteilt werden, so dass die Texte einer Kategorie inhaltlich zueinander passen. Anzahl und Charakterisierung der Kategorien sind dabei nicht vorgegeben, sondern sollen durch Gruppierung der Texte (*Clustering*) gefunden werden. Dies wird auch als *Document Clustering* bezeichnet. Es werden unterschiedliche Verfahren dazu eingesetzt; kurze Übersichten und weiterführende Literatur finden sich beispielsweise in [Schewe 1997], [Steinbach et al. 2000] oder [Zamir und Etzioni 1998].

Textklassifikation Dabei werden Klassen zur Einteilung der Texte vorgegeben und neue Texte jeweils einer Klasse zugeordnet. Die Zuordnung wird meistens aus klassifizierten Beispielen gelernt. Die Suche nach Texten zu einem bestimmten Thema könnte sich dann auf die Texte einer Klasse beschränken. Mit diesem Gebiet befasst sich Abschnitt 3.3.

Die Begriffe „Textklassifikation“ und „Textkategorisierung“ werden oft nicht unterschieden, dienen hier aber der Verdeutlichung (vgl. die Benutzung von *Categorization* in [Yang und Liu 1999]). Ich werde sie so benutzen wie hier definiert.

Die folgenden Überlegungen sollen deutlich machen, dass die obigen Gebiete viele Gemeinsamkeiten und Überschneidungen haben oder in Kombination verwendet werden können.

Zunächst benötigen alle Verfahren eine Repräsentation von Texten (und Benutzeranfragen), die inhaltliche Vergleiche ermöglicht. Aus Effizienzgründen kann man dabei nicht volle Sprachverarbeitung betreiben, wie das vorige Kapitel erläutert. Abschnitt 3.2 erläutert deshalb eine andere, inhaltlich weniger genaue, aber effizient zu erstellende Repräsentation. Weiterhin ist für jedes Gebiet ein hoher Aufwand nötig, um die Systeme so einzustellen, dass möglichst optimale Ergebnisse erzielt werden; dies legt den Einsatz von maschinellem Lernen nahe oder setzt ihn manchmal sogar voraus (für Beispiele siehe [Chen 1995] (IR), [McElligott und Sorensen 1994] (IF) oder [Yang 1999] (Textklassifikation) sowie die weiteren Abschnitte dieses Kapitels).

Information Retrieval und Information Filtering können auch als Klassifikation von Texten in „relevant“ und „nichtrelevant“ gesehen werden, wobei die Charakterisierung der Klasse „relevant“ als Benutzeranfrage (IR) oder Interessensprofil (IF) vorliegt. Bei der eigentlichen Textklassifikation wird dagegen eine Charakterisierung der Klassen meistens nicht vorgegeben, sondern aus Beispielen automatisch gelernt.

Beim Information Retrieval ist auch die Reihenfolge der Ausgabe der gefundenen Dokumente interessant. Es ist naheliegend, die Ähnlichkeit zur Anfrage zur Anordnung heranzuziehen. Wenn das System aber sehr viele Dokumente für relevant hält, kann es dem Benutzer helfen, sich zurechtzufinden, indem es diese Dokumente inhaltlich passend gruppiert. Dazu wird Textkategorisierung verwendet; vgl. beispielsweise [Hearst und Pedersen 1996].

Bei der Zusammenfassung von Texten unterscheidet man zwischen *Abstracts*, die den Textinhalt mit eigenen Formulierungen wiedergeben, und *Extracts* (Extrakten), die nur wichtige Ausschnitte des Ursprungstextes als Zusammenfassung präsentieren (siehe Abschnitt 3.5). Die Auswahl der richtigen Ausschnitte kann wieder als Klassifikation von Textteilen (Abschnitten oder Sätzen) in relevant oder nicht aufgefasst werden (*passage retrieval*), wie [Riloff 1993] anmerkt. Daher wird die Klassifikation von Texten und Textteilen in Abschnitt 3.3 genauer behandelt. Analog muss man bei der Informationsextraktion entscheiden, welche Textteile die gesuchte Information enthalten und welche nicht. Ebenso gibt es auch Verfahren zur Zusammenfassung, die nur nach bestimmten Inhalten suchen und nur diese zusammenfassen, was auch eine Art von Informationsextraktion ist, wie in der Einleitung erwähnt wird. Weitere Erläuterungen dazu folgen in Abschnitt 3.5. Andererseits kann Informationsextraktion zur Textklassifikation benutzt werden ([Riloff und Lehnert 1994]) und ebenso bei anderen Aufgaben helfen.

Mit diesen verschiedenen Sichtweisen ähnlicher Aufgaben wird auch klar, warum in der Literatur einige Begriffe nicht konsistent verwendet werden oder hier nicht genannte hinzukommen. Obwohl sehr vielfältige Aufgaben aus dem Bereich Text Mining bearbeitet worden sind und hier nicht alle Einsatzmöglichkeiten genannt werden können, lassen sich doch fast alle Ansätze mit der obigen Einteilung flexibel erfassen. Die Verkürzung von Emails zu SMS-Nachrichten beispielsweise, wie sie diese Diplomarbeit durchführt, wählt nur bestimmte Emails aus, wie beim Information Filtering, und verkürzt diese, ähnlich der Textzusammenfassung. Da nur bestimmte Informationen im verkürzten Text übrig bleiben, kann dies auch als Informationsextraktion betrachtet werden.

3.2. Repräsentation von Texten

Die im vorigen Abschnitt genannten Verfahren haben gemeinsam, Texte *inhaltlich* vergleichen zu müssen, ohne dass eine volle sprachliche und semantische Verarbeitung möglich ist. Die Texte werden daher in geeignete interne Repräsentationen überführt. Dieser Abschnitt bietet eine kurze Übersicht dazu und geht insbesondere auf die Repräsentation als *Wortvektor* ein, die in dieser Arbeit verwendet wird.

Zur Repräsentation eines Textes dienen *Indexterme*. Man kann zwischen zeichenketten- und wortbasierten Ansätzen unterscheiden. Erstere verwenden kein Wissen über Sprache und sind damit sprach- und domänenunabhängig. Das bekannteste Verfahren benutzt n-Gramme, also Zeichenketten, die durch ein über den Text geschobenes Fenster der Länge n (in Zeichen) gefiltert werden. Die 4-Gramme der Überschrift dieses Abschnittes lauten beispielsweise `Repr`, `eprä`, `präs` usw. Ein Text kann dann repräsentiert werden als Vektor, der für jedes n-Gramm die Anzahl seiner Vorkommen im Text angibt. Jedes n-Gramm ist also ein Indexterm. Die Vorteile dieser Verfahren sind ihre Geschwindigkeit, ihre Robustheit gegenüber Schreibfehlern und ihre Unabhängigkeit von Sprache und Domäne. Sie wurden zu verschiedenen Zwecken erfolgreich eingesetzt ([Damashek 1995], [Cohen 1995]). Ein Nachteil ist ihre Ungenauigkeit, die durch das Zerhacken von Wörtern entsteht. Da es bei den üblichen kleinen Werten von n mehr n-Gramme als Wörter in einem Text gibt, werden die Vektoren außerdem sehr lang.

Wortbasierte Ansätze verwenden Wörter als Indexterme. Sie müssen in der Lage sein, Wortgrenzen in einem Text zu erkennen (vgl. Abschnitt 2.1); dazu ist Sprachwissen im Allgemeinen nicht notwendig, aber hilfreich und bei Sprachen wie Chinesisch, deren Wortgrenzen nicht durch Zwischenräume erkennbar sind, auch notwendig. Als zweiten

Schritt nach Erkennung der Wortgrenzen nimmt man oft eine Stammformenreduktion (siehe Abschnitt 2.1) hinzu. Damit wird verhindert, dass verschiedene Formen desselben Wortes unterschiedlich repräsentiert werden. Dies bietet sich besonders für flektionsreiche Sprachen wie Deutsch an.

Auch bei den wortbasierten Ansätzen ist die Textdarstellung durch einen Vektor, der für jedes Wort seine Häufigkeit im Text angibt, naheliegend. Der Vektor hat dabei so viele Stellen, wie es Worte im ganzen Korpus (Textsammlung) gibt, damit alle Texte vergleichbar sind ([Salton und Buckley 1988]); für jeden Text sind also viele Stellen mit 0 belegt. Die Reihenfolge, in der die Wörter im Text vorkamen, wird durch die Vektordarstellung vernachlässigt, obwohl sie sicherlich nicht ohne Bedeutung für den Inhalt des Textes ist. Dies ist der sogenannte *Bag of Words*-Ansatz. Versuche, die Reihenfolge zu berücksichtigen, benutzen mehrwortige Phrasen (sie gehen also über den wortbasierten Ansatz hinaus), haben jedoch selten zu Verbesserungen der Ergebnisse geführt ([Lewis 1992]). Sie benötigen zudem eine aufwendigere Verarbeitung der Texte. Der Bag of Words-Ansatz scheint ein guter Kompromiss zwischen Aufwand und Qualität des Ergebnisses zu sein.

Allerdings sind die häufigsten Wörter die sogenannten *Stoppwörter*, also Funktionswörter wie Artikel, Pronomen oder Präpositionen, die in jedem Text vorkommen und nichts über seinen Inhalt aussagen. Oft werden diese Wörter deshalb vorher mit Hilfe einer Stoppwortliste entfernt; diese ist sprachabhängig. Alternativ kann man, wenn ein Lexikon zur Verfügung steht (Abschnitt 2.1), die Wörter mit solchen Wortarten aus dem Text entfernen. Die Zahl der Indexterme reduziert sich dadurch aber nicht wesentlich, da es nicht viele solcher Wörter gibt[2]. Andererseits können auch diese Wörter überraschend wichtig sein für bestimmte Aufgaben, wie in [Riloff 1997] dargestellt ist. Die Verwendung der Häufigkeit in der Vektordarstellung ist nur eine Annäherung an das eigentlich Erwünschte, nämlich eine Angabe, wie *charakteristisch* ein Wort für einen Text ist. Eine bessere Annäherung an diesen „Wert“ ist das *Tf-Idf*-Maß ([Salton 1989]). Es beruht auf der Annahme, dass für einen Text charakteristische Wörter in diesem Text häufig, in anderen Texten aber weniger häufig sind. Dieses Maß macht also das Gewicht eines Wortes nicht nur von seinem Text, sondern auch von den anderen Texten des Korpus und deren Charakteristik abhängig. Es ist wie folgt definiert:

Seien N Texte gegeben, und sei d_i der i-te Text. Sei t_k der k-te Indexterm in d_i. Dazu sei tf_{ik} die Häufigkeit (*term frequency*) von t_k in d_i, und n_k die Anzahl der Texte, in denen t_k mindestens einmal vorkommt. Dann ist w_{ik} das *Tf-Idf*-Gewicht von t_k in d_i und berechnet sich zu

$$w_{ik} = tf_{ik} * \log_2 \left(\frac{N}{n_k} \right) .$$

Mit tf_{ik} wird die Häufigkeit des Wortes im aktuellen Text berücksichtigt. Der zweite Teil der Gleichung ist die inverse Dokumenthäufigkeit (*inverse document frequency*) idf_k des Wortes t_k (daher die Bezeichnung *Tf-Idf*). Diese ist hoch, wenn das Wort in wenigen Texten (Dokumenten) vorkommt, und umgekehrt. Insgesamt ist der Wert also am höchsten, wenn das Wort häufig in seinem Text, aber selten in anderen Texten ist. Es sei

[2]Die Verteilung verschiedener Wörter in einem Korpus folgt annähernd Zipfs Gesetz: Es gibt wenige häufige Wörter und sehr viele, die nur selten auftauchen ([Joachims 2001]).

betont, dass das gleiche Wort in verschiedenen Texten verschiedene Gewichte bekommt, da es nicht für jeden Text gleich charakteristisch ist.

Bevor nun verschiedene Texte miteinander oder Texte mit Anfragen verglichen werden, werden die Dokumentvektoren zumeist auf die euklidische Länge eins normiert. Damit liegt eine von der Länge des Ausgangstextes unabhängige, einheitliche Repräsentation aller Texte der Sammlung vor. Auf dieser können nun Ähnlichkeitsfunktionen angewandt werden, um Texte miteinander oder mit einer Benutzeranfrage zu vergleichen. Ein weit verbreitetes Ähnlichkeitsmaß misst den Cosinus c des Winkels zwischen zwei Vektoren $\vec{d_i}$ und $\vec{d_j}$:

$$c = \frac{\langle \vec{d_i}, \vec{d_j} \rangle}{\|\vec{d_i}\| \cdot \|\vec{d_j}\|}$$

Hierbei bezeichnet „$\langle \cdot\,, \cdot \rangle$“ das Skalarprodukt. Wurden die Vektoren schon zur Länge eins normiert, so fällt der Nenner dieses Ausdrucks natürlich weg. Ein kleiner Wert von c entspricht einem großen Winkel zwischen den Vektoren und daher geringer Ähnlichkeit, und umgekehrt.

Die hier ausführlicher vorgestellte wortbasierte Darstellung von Texten ist nicht so schnell und robust wie die zeichenkettenbasierte Darstellung, aber noch schnell und robust genug für praktische Anwendungen. Sie beruht meistens auf sprachabhängigen Vorverarbeitungen des Textes (Stammformenreduktion, Stoppwortentfernung). Da sie auf Wörtern beruht, wird die Semantik eines Textes besser repräsentiert als etwa bei n-Grammen, weshalb sie weit verbreitet ist und auch in dieser Arbeit verwendet wurde. Allerdings erfasst sie Probleme wie Synonymie und Polysemie nicht, also die gleiche Bedeutung verschiedener Wörter beziehungsweise die verschiedenen Bedeutungen gleichgeschriebener Wörter. Um dies zu berücksichtigen, braucht man Thesauri, also Lexika, die die Bedeutungen der Wörter in Beziehung setzen. Mit diesem Zusatzaufwand kann man Bedeutungen als Indexterme verwenden. Gerade bei sehr kurzen Texten kann man so die Vergleichbarkeit verbessern, siehe beispielsweise [Burke et al. 1997]; bei normalen Texten kann man eher von einer Verwischung der Begriffe durch viele Nebenbedeutungen ausgehen, was sich nachteilhaft auswirkt.

3.3. Textklassifikation

Wie in der Übersicht schon erläutert wurde, kann die Klassifikation von Text*teilen* bei der gezielten Suche nach Informationen in Texten hilfreich sein. Deshalb wurde im Rahmen dieser Diplomarbeit mit einigen Textklassifikationsmethoden die Klassifikation von Sätzen, also sehr kurzen Texten, getestet. Abschnitt 3.3.3 bietet daher einige Überlegungen zu Besonderheiten der Klassifikation sehr kurzer Texte.

Eine andere Möglichkeit des Vorgehens ist, nur in Texten gezielt zu suchen, die vorher insgesamt als relevant klassifiziert wurden. Gerade wenn nur wenige Texte der Sammlung relevant für das gewünschte Thema sind, mag eine vorherige Klassifikation Arbeit sparen, wenn sie effizient durchgeführt werden kann. Auch dies wurde in dieser Arbeit getestet (vgl. Abschnitte 4.3 und 6.2.3). Dieser Abschnitt stellt daher die verwendeten Textklassifikationsmethoden vor.

Die meisten Methoden zur Textklassifikation setzen maschinelles Lernen ein, da die manuelle Erstellung von Regeln sehr aufwendig werden kann, gerade bei vielen möglichen Klassen oder Veränderungen der gewünschten Klassen. Das Lernszenario kann wie folgt formalisiert werden ([JOACHIMS 2001]): Gegeben ist eine Trainingsmenge S von N Beispieltexten, repräsentiert durch ihre Indextermvektoren (siehe vorigen Abschnitt) $\vec{x}_1, \ldots, \vec{x}_N$ aus dem Vektorraum $X = \mathbb{R}^n$. Jedem Vektor aus X ist eine Klasse y aus einer Menge Y zugeordnet:

$$S = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_N, y_N)\}$$

Im einfachsten Fall einer binären Klassifikation (zum Beispiel in „relevant" und „nicht relevant") kann man etwa $Y = \{1, -1\}$ wählen. Es wird angenommen, dass die Zusammensetzung der Trainingsmenge der unbekannten Verteilung $\Pr(\vec{x}, y)$, mit der die Klassen und Texte vorkommen, entspricht. Diese Verteilung gibt die Lernaufgabe vor: Gelernt werden soll eine Funktion $h : X \longrightarrow Y$, die beliebigen Texten eine Klasse aus Y zuordnet und dabei (nach einem zu spezifizierenden Fehlermaß, dem *Verlust*) möglichst wenig Fehler macht. Die Funktion h wird auch Hypothese genannt und stammt aus einem Hypothesenraum H. Eine naheliegende Verlustfunktion ist

$$L(h(\vec{x}), y) = \begin{cases} 0 & h(\vec{x}) = y \\ 1 & sonst \end{cases}$$

Die Performanz eines Klassifizierers, seine Fehlerrate $Err(h)$, ist die Wahrscheinlichkeit, dass auf einem zufällig nach $\Pr(\vec{x}, y)$ gezogenen Beispiel eine falsche Vorhersage gemacht wird:

$$Err(h) = \Pr(h(\vec{x}) \neq y \,|\, h) = \int L(h(\vec{x}), y) \, d\Pr(\vec{x}, y)$$

Wie aus [JOACHIMS 2001] und [YANG und LIU 1999] hervorgeht, sind Support Vector Machines (SVMs) eine der besten Lernmethoden zur Textklassifikation (weitere bekannte Verfahren werden dort ebenfalls vorgestellt). SVMs kommen daher in dieser Arbeit zum Einsatz und werden im folgenden Unterabschnitt vorgestellt. Eine sehr einfache Methode, für die aber gute Ergebnisse berichtet worden sind ([HAN und KARYPIS 2000]), verwendet Zentroide der Dokumentvektoren einer Klasse, wie Abschnitt 3.3.2 erläutert.

3.3.1. Support Vector Machines

Support Vector Machines (SVMs, [CORTES und VAPNIK 1995]) sind eine Entwicklung aus dem Bereich der statistischen Lerntheorie. Sie beruhen auf der Idee der *strukturellen Risikominimierung* ([VAPNIK 1982]) und wurden erstmals in [JOACHIMS 1998] für Textklassifikation eingesetzt. In [JOACHIMS 2001] wird ihr Einsatz für Textklassifikation auf eine theoretische Grundlage gestellt. Die folgenden Darstellungen stützen sich wesentlich auf diese Arbeit, sowie auf [BURGES 1998].

Strukturelle Risikominimierung beruht auf der Tatsache, dass die Fehlerrate $Err(h)$ einer Hypothese h aus H mit der Komplexität von H und der Trainingsfehlerrate $Err_{tr}(h)$ in Verbindung gebracht werden kann. Die Komplexität von H wird als VC-Dimension d angegeben. Sie ist definiert als die maximale Anzahl von Beispielen, die eine Funktion aus H bei beliebiger Klassifikation der Beispiele korrekt trennen kann. Als anschauliches Beispiel betrachte man die reelle Ebene $\mathbb{R}^2$ und drei linear unabhängige Punkte darin, sowie für H die Klasse der Geraden. Wie man sich leicht klarmacht, gibt es für jede binäre Klassifikation der drei Punkte eine Gerade, die so zwischen den Punkten liegt, dass sich alle positiv klassifizierten Beispiele auf einer Seite und alle negativen auf der anderen befinden. Für vier Punkte geht dies nicht, die VC-Dimension von H als Klasse der Geraden im $\mathbb{R}^2$ ist also $d = 3$. Dies lässt sich verallgemeinern: die VC-Dimension von Hyperebenen aus dem $\mathbb{R}^n$ beträgt $n+1$. Anders ausgedrückt, haben lineare Schwellwertfunktionen mit n Attributen die VC-Dimension $n + 1$ ([Vapnik 1998]).

Die folgende Schranke ([Vapnik 1998]) leistet die Verbindung von Fehlerrate und Komplexität d der Hypothese h. Dabei ist N wie oben die Anzahl der Trainingsbeispiele und $1 - \eta$ die Wahrscheinlichkeit, mit der die Schranke gilt.

$$Err(h) \leq Err_{tr}(h) + O\left(\frac{d \ln\left(\frac{N}{d}\right) - \ln(\eta)}{N}\right) \tag{3.1}$$

Der echte Fehler $Err(h)$ ist also abhängig einerseits vom Trainingsfehler und andererseits von der Komplexität der verwendeten Lernfunktionen. Anschaulich formuliert, liefern einfache Funktionen in den meisten Fällen keinen guten Trainingsfehler, da sie nicht gut trennen können. Andererseits ergeben sehr komplexe Funktionen gute Trainingsfehler, aber einen hohen Wert für den rechten Teil der obigen Schranke. Dies kann man so interpretieren, dass die Funktion nur die Trainingsmenge gut trennt, aber keine gute Vorhersagefähigkeit für weitere Punkte hat; man spricht von *Overfitting*. In beiden Fällen ist die Schranke lose. Die Wahl des richtigen Hypothesenraumes ist also ausschlaggebend.

Bei struktureller Risikominimierung wählt man daher eine Struktur von ineinander verschachtelten Hypothesenräumen mit zunehmender Komplexität:

$$H_1 \subset H_2 \subset \ldots \subset H_i \subset \ldots \quad \text{wobei} \quad \forall i : d_i \leq d_{i+1}$$

Damit gilt es, den Index i auszuwählen, so dass Gleichung 3.1 minimiert wird. Wie oben erläutert wurde, hängt die Komplexität des Hypothesenraumes von der Anzahl der Attribute ab. Ein naheliegendes Vorgehen ist also, die Anzahl der Attribute schrittweise zu erhöhen und damit immer komplexere Räume zu untersuchen. Dabei sollten die „wichtigsten“ Attribute zuerst getestet werden—aber welche das sind, ist nicht a priori bekannt. Bei Textklassifikation ist dieses Vorgehen schon deshalb nicht möglich, weil die Anzahl der Attribute der Anzahl der Wörter im Korpus entspricht (siehe Abschnitt 3.2), also sehr hoch ist.

Support Vector Machines reduzieren daher die Komplexität des Hypothesenraumes auf andere Weise. Generell finden sie eine Hyperebene, die die durch Vektoren gegebenen Punkte richtig trennt; es sei für die Darstellung hier vorausgesetzt, dass eine Trennung der gegebenen Punkte tatsächlich möglich ist, was bei Textklassifikation meistens erfüllt ist, da die Indextermvektoren an vielen Stellen 0 enthalten. Für nicht trennbare Punktmengen wird das Modell erweitert und führt zu einem ähnlichen Optimierungsproblem.

Im Allgemeinen gibt es mehrere mögliche Hyperebenen, die die Trainingspunkte richtig trennen; SVMs wählen davon genau diejenige aus, die den Abstand zu den nächstgelegenen Punkten maximiert (siehe Abb. 3.1). Dieser Abstand ist der sogenannte *Margin*. Der Grund für die Maximierung des Margins ist, dass hoher Margin einer niedrigen VC-Dimension, also niedriger Komplexität, der Hyperebene entspricht ([Vapnik 1982]). Dies

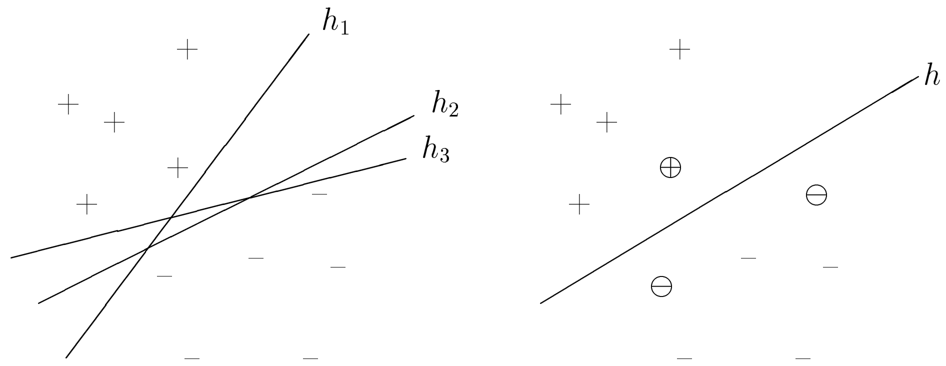

Abbildung 3.1.: Veranschaulichung der Trennung von Punkten durch Hyperebenen in der zweidimensionalen Ebene. Links mehrere trennende Geraden, rechts die Gerade mit maximalem Abstand zu den nächstgelegenen Punkten (Support Vectors), im Bild eingekreist.

wird wie folgt formalisiert: Die zu findende Hyperebene ist von der Form $\vec{w} \cdot \vec{x} + b = 0$ mit Normalvektor $\vec{w}$ und Abstand vom Ursprung $b/\|\vec{w}\|$. Gesucht ist also die Nullstelle einer Funktion f mit $f(\vec{x}) = \vec{w} \cdot \vec{x} + b$, wobei alle Trainingspunkte $\vec{x}_i$ richtig getrennt werden:

$$y_i(\vec{w} \cdot \vec{x}_i + b) > 0 \qquad \forall i = 1, \ldots, N$$

Die Funktion f ist damit aber noch nicht eindeutig festgelegt. Mit der zusätzlichen Forderung

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 \qquad \forall i = 1, \ldots, N \tag{3.2}$$

wird die Funktion festgelegt und ein gewisser Mindestabstand δ der nächstgelegenen Punkte zur Ebene erzwungen. Diese Punkte heißen *Support Vectors* und definieren den Margin δ. Sie allein bestimmen die Lage der trennenden Hyperebene. Sie liegen auf Hyperebenen, die zu der ersten parallel liegen und wegen (3.2) in der Form

$$\begin{array}{ll} \vec{w} \cdot \vec{x}_i + b = 1 & \text{für } y_i = 1 \\ \vec{w} \cdot \vec{x}_i + b = -1 & \text{für } y_i = -1 \end{array}$$

angegeben werden können. Der Abstand vom Ursprung dieser parallelen Ebenen beträgt also $|1 - b| \,/\, \|\vec{w}\|$ bzw. $|-1 - b| \,/\, \|\vec{w}\|$, damit berechnet sich ihr jeweiliger Abstand zur ersten Ebene, der Margin, zu $\delta = 1 \,/\, \|\vec{w}\|$. Man maximiert also den Margin, wenn man $\|\vec{w}\|$ minimiert.

Die Verbindung von hohem Margin mit niedriger VC-Dimension wird durch folgendes Lemma erreicht ([Vapnik 1982], zitiert nach [Joachims 2001]): Seien alle Beispielvektoren x_i in einer Kugel mit Radius R enthalten und gelte für sie $|\vec{w} \cdot \vec{x}_i + b| \geq 1$. Dann hat die Menge der Hyperebenen $h(\vec{x}) = sign\{\vec{w} \cdot \vec{x} + b\}$ im $\mathbb{R}^n$, aufgefasst als Hypothesen, eine VC-Dimension d, die beschränkt ist durch

$$d \leq min(\left\lceil \frac{R^2}{\vec{w}^2} \right\rceil, n) + 1\,.$$

Hier ist also die VC-Dimension nicht mehr abhängig von der Zahl der Attribute, sondern von $\|\vec{w}\|$, der Euklidischen Länge des Normalenvektors $\vec{w}$ der trennenden Hyperebene. Zusammengefasst, müssen Support Vector Machines also folgendes Optimierungsproblem lösen:

$$\begin{array}{ll} \text{Minimiere:} & V(\vec{w}, b) = \|\vec{w}\| \\ \text{so dass gilt:} & y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 \,, \; \forall i = 1, \ldots, N \end{array}$$

Da dieses quadratische Optimierungsproblem numerisch schwierig ist, wird eine duale Form dazu verwendet, die auch für Probleme mit vielen Attributen und vielen Trainingsbeispielen effizient gelöst werden kann; Details dazu finden sich in [Joachims 2001]. Für das Verständnis des Einsatzes von SVMs im Rahmen dieser Arbeit reichen die bisherigen Darstellungen aus. Support Vector Machines finden effizient für eine gegebene Menge klassifizierter Punkte im $\mathbb{R}^n$ diejenige Hyperebene, die sie richtig trennt und den größten Abstand zu den nächstgelegenen Punkten hat.

Zur Vorhersage der Klasse neuer Texte wird mit deren Vektordarstellung $\vec{x}$ berechnet, auf welcher Seite der Hyperebene sie liegen:

$$y = sign(\vec{w} \cdot \vec{x} + b) \,.$$

Der Einsatz von Support Vector Machines im Rahmen dieser Arbeit wird im Kapitel 4 erläutert.

3.3.2. Zentroidbasierte Klassifikation

Diese einfache Klassifikationsmethode wurde hinzugenommen, weil sie auf der gleichen Vektordarstellung von Texten beruht wie die SVM, so dass der Aufwand für ihre Implementierung sehr gering war. Außerdem werden in [Han und Karypis 2000] gute Ergebnisse für sie berichtet: sie schnitt bei den dortigen Experimenten auf verschiedenen Daten im Vergleich zu den klassischen Textklassifikationsmethoden *Naive Bayes* (siehe bspw. [Mitchell 1997]), *C4.5* ([Quinlan 1993]) und *k-Nearest Neighbor* (etwa [Iwayama und Tokunaga 1995]) am besten ab.

Für die Klasse $y \in Y$ sei S_y die Menge der Indextermvektoren aus der Trainingsmenge, die dieser Klasse zugeordnet sind. Dann berechnet sich ihr Zentroidvektor $\vec{c}_y$ zu

$$\vec{c}_y = \frac{1}{|S_y|} \sum_{\vec{x} \in S_y} \vec{x} \,.$$

Der Zentroidvektor enthält also den Mittelwert der Gewichte aller Indexterme aus den Texten mit der gleichen Klasse. Er hat im Allgemeinen nicht die Länge eins, auch wenn die Indextermvektoren auf Länge eins normiert wurden.

In der Lernphase wird für jede Klasse ihr Zentroidvektor auf Basis der Trainingsmenge berechnet. Die Trainingszeit ist also linear in der Anzahl der Texte und Indexterme. Ein neu zu klassifizierender Text wird dann mit allen Zentroiden verglichen, im Falle dieser Diplomarbeit und [Han und Karypis 2000] über den Winkel wie in Abschnitt 3.2 beschrieben. Der ähnlichste Zentroid bestimmt die Klasse des neuen Textes. Es handelt sich um eine sehr einfache und effiziente Methode.

3.3.3. Satzklassifikation

Die Klassifikation von Texten bietet für Lernverfahren einige besondere Herausforderungen, die in diesem Abschnitt kurz erläutert werden. Es geht dabei um die Frage, ob die Klassifikation von Sätzen als sehr kurzen Texten mit denselben Verfahren wie für längere Texte erfolgversprechend ist.

Nach [Joachims 2001] lässt sich Textklassifikation im Hinblick auf Lernverfahren durch folgende Aspekte charakterisieren:

Hochdimensionaler Datenraum Durch die Darstellung von Texten mit Indextermvektoren, wobei zumeist jede Stelle des Vektors einem möglichen Wort entspricht, ist die Anzahl der Attribute für Lernverfahren sehr hoch.

Dünnbesetzte Attributvektoren In jedem einzelnen Text kommt nur ein kleiner Teil aller Wörter des Korpus vor, so dass die Vektoreinträge eines Indexvektors fast überall 0 sind.

Hohe Redundanz Die meisten Texte enthalten mehrere Wörter, die Hinweise auf ihre Klasse geben.

Heterogene Wortverteilung Normalerweise ist es nicht so, dass ein Wort oder eine kleine Gruppe von Wörtern genügt, um anhand ihrer die Klasse beliebiger Texte zu bestimmen. In natürlicher Sprache gibt es immer verschiedene Möglichkeiten, einen Sachverhalt auszudrücken, so dass die Überschneidung in der Wortwahl zwischen zwei inhaltlich ähnlichen Texten sehr gering sein kann. Im Extremfall gibt es keine Überschneidung. Hat man jedoch eine Gruppe von inhaltlich ähnlichen Texten, so wird es meist eine gewisse Menge von Wörtern geben, mit der jeder Text der Gruppe eine Überschneidung aufweist.

Rauschen Auch redaktionell bearbeitete Texte sind nicht frei von Rechtschreib- und Tippfehlern oder ungrammatischen Sätzen. Umso mehr gilt dies für frei geschriebene Texte wie zum Beispiel Emails.

Während die ersten beiden und der letzte Aspekt die Lernaufgabe erschweren, wirkt hohe Redundanz nachweislich erleichternd ([Joachims 2001], darin Abschnitt 4.4.3). Bei sehr kurzen Texten (Sätzen) verschärft sich der zweite Punkt noch, weil nur wenige Wörter in einem Satz enthalten sind. Andererseits ist aus dem selben Grund mit weniger Redundanz innerhalb eines Satzes zu rechnen. Satzklassifikation scheint daher schwieriger zu sein als Textklassifikation. Zwar kann das Niveau der Redundanz in der ganzen Satzsammlung vergleichbar mit einer Sammlung von längeren Texten sein, wenn eine genügend große Gruppe von Wörtern zusammen Hinweise auf die Klasse eines Satzes gibt. Die Menge der Überschneidungen jedes Satzes mit dieser Gruppe ist aber, aufgrund der Kürze des Satzes, wieder geringer als bei längeren Texten.

Aus diesen Gründen kann angenommen werden, dass die Klassifikation von Sätzen mit bekannten Textklassifikationsmethoden nicht so gute Performanz erbringt wie die von längeren Texten. Mehr Erfolg verspricht gezieltes Satzfiltern in Texten, die vorher insgesamt als relevant klassifiziert wurden. Abschnitt 6.2.3 nennt die Ergebnisse der experimentellen Überprüfung dieser Mutmaßungen.

3.4. Informationsextraktion

Die Verfahren zur Informationsextraktion bilden insofern eine Ausnahme in diesem Kapitel, als bei ihnen die Repräsentation der Texte nicht soweit abstrahiert ist, wie dies in den anderen genannten Bereichen oft der Fall ist. Insbesondere wird keine Vektordarstellung verwendet. Der Grund ist, dass die extrahierten Informationen in für Menschen verständlicher Form vorliegen müssen. Bei den anderen Verfahren geht es eher um eine maschinelle Auswahl von Texten oder Textteilen als um Operationen mit dem Text oder einzelnen Sätzen, wie bei der Informationsextraktion. Informationsextraktion arbeitet also „näher am Text“. Dies gilt auch für die automatische Textzusammenfassung (siehe Abschnitt 3.5.1).

Informationsextraktion steht in der Nachfolge früherer Systeme, die sich mit vollem Textverständnis befassten und die unter großem Aufwand erstellt wurden, aber schlecht mit beliebigen Texten umgehen konnten und schwierig zu bewerten waren (wann ist ein Text „verstanden worden“?). Mit den Message Understanding Conferences (MUC) ab Ende der achtziger Jahre entwickelte sich daher eine reduzierte Aufgabe[3]: Es wird nur nach bestimmten, vorher festgelegten Informationen gesucht. Die Festlegung erfolgt durch spezifische Ausgabeformate, sogenannte *Templates* oder Schablonen, die eine Anzahl von *Slots* (Steckplätzen) aufweisen, die gefüllt werden mit einzelnen Phrasen oder Wörtern des Textes ([Appelt und Israel 1999], [Glickman und Jones 1999], [Cowie und Lehnert 1996]). Beispielsweise ging es bei der siebten und bisher letzten MUC um Zeitungsartikel, die Starts von Raketen und Raumschiffen zum Inhalt hatten. Aufgabe der an der Konferenz teilnehmenden Systeme war es, jeden im Korpus erwähnten Start zu erkennen und mit Detailangaben zu Art des Raumschiffes, Nutzlast, Startzeitpunkt und Startort zu versehen. Dazu sollte erkannt werden, ob der Start gelungen war und ob es sich um einen zivilen oder militärischen Flug handelte. Hier gab es also nur ein mögliches Template (Start eines Raumschiffes/einer Rakete). Templates können sich für eine weitere maschinelle Verarbeitung eignen, beispielsweise in partiell strukturierten Datenbanken.

Ein typisches Informationsextraktionssystem benutzt, um den Text zu analysieren, die in Kapitel 2 angeführten sprachverarbeitenden Verfahren bis hin zu flachen Parsern ([Appelt und Israel 1999]). Zusätzliche Elemente sind eine Eigennamenerkennung, mit der Namen von Personen, Firmen etc. erkannt werden, und oft ein Koreferenz-Modul, mit dem versucht wird, verschiedene Erwähnungen desselben Objektes zusammen zu bringen. Zum Beispiel könnte eine Firma zunächst mit ihrem Namen und später mit ihrem Firmensitz oder dem Namen des Eigentümers genannt werden. Eine Beschreibung der Methoden zur Erkennung von Koreferenz ist hier nicht notwendig (sie findet sich aber in [Appelt und Israel 1999]).

Zur Erkennung der relevanten Textstellen benutzen solche Systeme Patterns, die genau angeben, welche Wörter (oder Wortarten) in welcher Reihenfolge gefunden werden müssen, um die Wahl eines Templates zu rechtfertigen (*Triggern* eines Templates) oder einen Slot zu füllen. Außerdem verwenden sie syntaktische Angaben des Parsers und/oder andere Ergebnisse der Sprachverarbeitung. Eine Übersicht über solche Patterns findet sich in [Muslea 1999]. Da es viele Möglichkeiten gibt, einen Sachverhalt

[3]Vgl. das Lichtenberg-Zitat am Anfang dieser Diplomarbeit.

in natürlicher Sprache auszudrücken, sind viele Patterns pro Template notwendig und die Zusammenstellung bedeutet einen sehr hohen Aufwand. Deshalb wird hier auch maschinelles Lernen eingesetzt; dazu gibt [Glickman und Jones 1999] eine gute Übersicht. Nach [Appelt und Israel 1999] benötigen allerdings auch Lernverfahren hohen menschlichen Aufwand, da die zum Lernen notwendige Vorkennzeichnung von Text mit dem für Informationsextraktion nötigen Detailliertheitsgrad mühsam und fehleranfällig ist. Der Ansatz in [Riloff und Jones 1999] bietet hier einen Ausweg, indem der Kontext von wenigen Ausgangswörtern einer semantischen Kategorie benutzt wird, um weitere Wörter der Kategorie zu finden, deren Kontexte schließlich (nach Iteration) zu Patterns führen. Menschlich erstellte Patterns liefern aber bisher etwas bessere Ergebnisse als automatisch gelernte ([Appelt und Israel 1999]).

Diese Übersicht über Methoden zur Informationsextraktion mag etwas knapp und oberflächlich sein, dient hier aber nur zur Einordnung der Diplomarbeit: keine speziellen Verfahren solcher Systeme kommen in dieser Arbeit zum Einsatz. Stattdessen sollen, wie der Titel der Arbeit andeutet, Informationen gefunden werden, indem Teile jedes Textes auf das Vorkommen der gewünschten Information geprüft werden, dann aber nicht weiter analysiert werden außer zur eventuellen Verkürzung (Kapitel 5). Ein ähnliches Vorgehen findet man bei der gezielten Textzusammenfassung, auf die im nächsten Abschnitt eingegangen wird.

3.5. Automatische Textzusammenfassung

Zusammenfassungen eines Textes verhelfen Lesern zu einem schnellen Überblick über den Inhalt des Textes, ohne ihn ganz lesen zu müssen. Menschlich erstellte Zusammenfassungen geben dazu idealtypisch die wesentlichen inhaltlichen Punkte in geraffter Form, mit eigenen Formulierungen und übersichtlich geordnet wieder. Damit verbunden ist eine Einordnung verschiedener Textteile im Hinblick auf ihre Relevanz. Passagen, die einen schon genannten Punkt näher ausführen oder Beispiele geben, werden in einer Zusammenfassung weniger berücksichtigt als einleitende oder zusammenfassende Abschnitte. Die äußere Struktur eines Textes spielt dabei genauso eine Rolle wie das menschliche Hintergrundwissen.

In [Hovy und Marcu 1998] werden Zusammenfassungen von Texten nach folgenden Gesichtspunkten eingeordnet:

- *Indikativ* oder *informativ*: Indikative Zusammenfassungen dienen nur zur schnellen Einordnung des Textes; informative Zusammenfassungen machen ein Verständnis des Textinhalts ohne Lesen des ganzen Textes möglich.

- *Extrakt* oder echte Zusammenfassung, englisch *Abstract*: Ein Extrakt enthält nur Teile des Originaltextes; ein Abstract ist neu formuliert nach inhaltlichem Verständnis.

- *Generisch* oder *gezielt*[4]: Generische Zusammenfassungen geben möglichst alle wichtigen Aspekte eines Textes wieder; gezielte oder anfrageorientierte Zusammenfassungen berücksichtigen nur Aspekte, die im Interesse des Benutzers stehen.

[4] Zum Begriff *gezielt* vgl. Seite 26.

- Einzelner Text oder Textsammlung: Eine Zusammenfassung kann nur für einen Text oder für mehrere (inhaltlich verwandte) Texte erstellt werden.

In [Hahn und Mani 1998] wird außerdem der Begriff des *Kondensats* verwendet, der eine formale Repräsentation des reduzierten Inhalts eines Textes bezeichnet. Weitere mögliche Aspekte zur Charakterisierung einer Zusammenfassung sind Textgenre, Domäne und die Frage, ob eine Bewertung oder Kritik des Textes enthalten ist.

Mit diesen Begriffen sind zum Beispiel die gängigen Zusammenfassungen, die zu Beginn wissenschaftlicher Arbeiten angegeben werden, informative generische Abstracts eines einzelnen Textes. Ihre automatische Erstellung ist mit großen Schwierigkeiten verbunden, da sie tiefes Textverständnis und damit viel Hintergrundwissen voraussetzt sowie eine Komponente zur Textgenerierung. In [Hahn und Mani 1998] werden einige Ansätze hierzu aufgeführt, die auf gewissen formalen, semantischen Repräsentationen von Textinhalten operieren und deren Kondensate zum Ausgangspunkt für Zusammenfassungen nehmen. Diese Autoren merken aber an, dass die Arbeiten entweder theoretisch geblieben sind oder die entstandenen Systeme schlecht skalieren, also für beliebige Textsammlungen ungeeignet sind; die Schwierigkeiten bei der Entwicklung textverstehender Systeme wurden schon im vorigen Abschnitt und in Kapitel 2 erwähnt.

Daher befassen sich viele Systeme zur Textzusammenfassung mit einfacheren Methoden, die zum Ziel haben, indikative Extrakte zu erstellen ([Goldstein et al. 1999]). Die Zusammenfassung ist also eine Zusammenstellung von Absätzen oder Sätzen des Ursprungstextes, die mit verschiedenen Methoden ausgewählt werden können. Die meisten Arbeiten verwenden eine satzweise Extraktion, auch Satzfiltern genannt; Absätze werden beispielsweise in [Mitra et al. 1997] und [Strzalkowski et al. 1998] extrahiert. Das Vorgehen kann als Klassifikation von Sätzen bzw. Absätzen als verwendbar für einen Extrakt oder nicht gesehen werden.

In dieser Diplomarbeit spielt Satzfiltern eine zentrale Rolle. Allerdings sollen nur Sätze zum gewünschten Thema extrahiert werden, was ich *gezieltes Satzfiltern* nennen möchte. Die nächsten beiden Unterabschnitte behandeln daher Verfahren zum Satzfiltern aus der Literatur und die Vorgabe von Themen bei automatischer Zusammenfassung. Es wird deutlich, dass die genannten Verfahren für die gestellte Aufgabe nicht geeignet sind oder noch angepasst werden müssen. Das nächste Kapitel erläutert daher den eigenen Ansatz zum gezielten Satzfiltern.

Vorher soll noch auf ein Problem von Extrakten hingewiesen werden: Da Sätze aus dem Zusammenhang gerissen werden, ist die Lesbarkeit der Extrakte natürlich nicht so gut wie die von neuformulierten Abstracts. Das größte Problem bei der Lesbarkeit ist die Verwendung von Anaphora, also Pronomen und anderen Ausdrücken, deren Bezug nur aus vorhergehenden Sätzen klar wird. Ein Satz wie „Vor jenem Ereignis ging für sie alles gut" verdeutlicht dies: Sowohl das Ereignis als auch die Person müssen in vorigen Sätzen erwähnt worden sein, um den Satz verständlich zu machen. Zwar existieren Verfahren, die zur Auflösung von anaphorischen Bezügen dienen, doch sind diese nur mäßig erfolgreich und setzen oft weitgehende sprachliche Verarbeitung der Texte voraus ([Lappin und Leass 1994], [Kennedy und Boguraev 1996]). Daher verzichtet man bei der Erstellung von Extrakten meist auf eine Behandlung des Problems, zumal Extrakte nur indikative Zusammenfassungen sein sollen.

Zur Textzusammenfassung existieren neben den erwähnten noch weitere Verfahren, die zum Beispiel auf der rhetorischen Struktur eines Textes beruhen ([Marcu 1998]) oder im Text vorkommende Konzepte durch Verwendung übergeordneter Begriffe verallgemeinern ([Lin 1995]). Auch Informationsextraktion im Sinne von Abschnitt 3.4 kann verwendet werden ([McKeown und Radev 1995]). Solche Verfahren spielen für diese Arbeit keine Rolle.

3.5.1. Satzfiltern

Mit Satzextraktion zur Zusammenfassung haben sich eine Reihe von Arbeiten beschäftigt. Generell gehen sie so vor, jedem Satz des Textes ein Gewicht zuzuweisen und die Sätze mit dem höchsten Gewicht für die Zusammenfassung zu verwenden. Dabei kann entweder die maximale Zahl der Sätze für jede Zusammenfassung vorgegeben werden (feste Extraktlänge) oder das Mindestgewicht für einen Satz, um extrahiert zu werden (variable Extraktlänge). Das Satzgewicht kann aus einer oder mehreren der folgenden Eigenschaften eines Satzes gewonnen werden (vgl. [Hovy und Marcu 1998]).

- Position des Satzes: Bei Zeitungsartikeln sind meistens die ersten Sätze schon die beste Zusammenfassung, weil Zeitungsartikel so angelegt sind, dass die wichtigsten Informationen zuerst genannt werden. Bei anderen Textgenres könnten andere Positionen wichtig sein. Die Überschrift eines Textes ist immer sehr wichtig.

- Schlüsselformulierungen (*Cue Phrases*): Sätze, die Formulierungen wie „In diesem Artikel geht es um ...“, „Zusammenfassend ...“ oder „Wichtig ist, ...“ enthalten, sind wichtiger als solche mit „Als Beispiel ...“ oder „Am Rande sei bemerkt ...“.

- Charakteristische Wörter: Sätze, die Wörter enthalten, die in diesem Text häufig sind (abgesehen von Stoppwörtern), oder die nach anderen statistischen Maßen (wie *Tf-Idf*, Abschnitt 3.2) für den Text charakteristisch sind, sind gut für Zusammenfassungen geeignet. Eine andere Methode, charakteristische Wörter zu erhalten, verwendet ihre semantische Verwandtschaft. Dies wird *Lexical Cohesion* genannt, siehe unten.

Bewertet wird Satzfiltern durch Vergleich mit menschlichen Vorgaben: In den meisten Tests wurden Testpersonen gebeten, für jeden ihnen vorgelegten Text die Sätze zu bestimmen, die für den Text am wichtigsten sind. Dies ist natürlich subjektiv, und die Übereinstimmung zwischen den Testpersonen ist nicht unbedingt hoch (vgl. [Rath et al. 1961]). Es fehlt eine Standardsammlung von Texten mit vorgegebenen Markierungen der wichtigsten Sätze, anhand derer man die verschiedenen Verfahren vergleichen könnte.

In [Lin und Hovy 1997] werden wichtige Satzpositionen mit Hilfe eines Trainingskorpus ermittelt. Im Korpus sind die zu extrahierenden Sätze markiert. Jeder Satz ist durch die Nummer seines Abschnittes und seine Nummer in diesem Abschnitt gekennzeichnet. Durch Bestimmung einer Ordnung über den möglichen Satzpositionen erhält man eine allgemeine Methode zum Auffinden relevanter Sätze nur über ihre Position. Die Ergebnisse deuten darauf hin, dass man dieses Verfahren nur in Kombination mit anderen verwenden sollte. Zum gezielten Satzfiltern ist es nicht geeignet.

Zum Standardverfahren hat sich die Gewichtung mittels *Tf-Idf* (Abschnitt 3.2) entwickelt ([Brandow et al. 1995], [Zechner 1996], [Buyukkokten et al. 2000]). Die *Tf-Idf*-Gewichte aller Wörter eines Satzes werden summiert, was direkt ein Ranking aller Sätze ergibt. In [Zechner 1996] werden Ergebnisse zwischen 55% Recall bei 46% Precision bis zu 91% Recall bei 37% Precision genannt, je nach Anzahl der Sätze, die zur Zusammenfassung zugelassen werden (Recall und Precision werden im folgenden Kapitel, Abschnitt 4.3 erläutert). Bei mehr möglichen Sätzen, also größerer Länge der Extrakte, steigt der Recallwert, während der Precisionwert sinkt. *Tf-Idf* ist ein Maß, das die Charakteristik eines Wortes für seinen ganzen Text annähert; damit ist diese Methode nicht direkt übertragbar für gezieltes Satzfiltern, wo man die Charakteristik eines Wortes für ein bestimmtes Thema benötigt. Kapitel 4 (Abschnitt 4.2.2) beschreibt, wie dieses Maß mit Anpassungen für gezieltes Satzfiltern verwendet werden kann.

In [Kupiec et al. 1995] wird Satzextraktion mit maschinellem Lernen verbunden, wozu eine Mischung verschiedener Eigenschaften (Attribute) von Sätzen benutzt wird. Diese Autoren verwenden Satzlänge, Satzposition innerhalb eines Absatzes, Vorkommen von im Text häufigen Worten und Vorkommen von Schlüsselformulierungen, um einen Satz zu charakterisieren. Als Lernverfahren benutzen sie den Naive-Bayes-Klassifikator (siehe bspw. [Mitchell 1997]). Dabei wurden die relevanten Sätze des Korpus nicht per Hand markiert, sondern durch Vergleich mit Sätzen, die in menschlich erstellten Zusammenfassungen der Texte vorkamen, für relevant befunden oder nicht. Das Verfahren lernt also nur aus schon erstellten Zusammenfassungen und erspart die satzweise Markierung des Korpus. Das beste Resultat sind 43% Recall und Precision für Satzextraktion. Die Worthäufigkeit erwies sich dabei als Attribut, dessen Verwendung das Lernergebnis verschlechterte.

Lernende Verfahren bieten sich auch für gezieltes Satzfiltern an, wenn es möglich ist, für gewünschte Themen einen Trainingskorpus zusammenzustellen. Dass Satzlänge und Satzposition aber gute Eigenschaften sind, um verschiedene Themen auseinander zu halten, darf bezweifelt werden. Auch die Schlüsselformulierungen können themenabhängig sein, sollten also selbst erst gelernt werden, damit nicht für jedes Thema eine Liste von Hand erstellt werden muss. Das eben vorgestellte Verfahren bietet sich also für gezieltes Satzfiltern nicht an. Ohnehin passt die Vorgabe von fertigen Zusammenfassungen nicht zur Aufgabenstellung dieser Diplomarbeit.

Ebenfalls aus vorhandenen Zusammenfassungen lernen [Mani und Bloedorn 1998] und berichten verbesserte Ergebnisse. Diese Autoren benutzen ebenfalls ein Ähnlichkeitsmaß, um die dem gegebenen Abstract (als Ganzem) ähnlichsten Sätze des Textes zu bestimmen. Diese Sätze werden als positive Beispiele, die anderen als negative Beispiele verstanden. Es wird eine Kombination verschiedener Attribute der Sätze zum Lernen verwendet: Attribute zur Satzposition, zu charakteristischen Wörtern und zu inhaltlichen Zusammenhängen von Wörtern (*Lexical Cohesion*, siehe nächsten Absatz). Es werden drei maschinelle Lernverfahren eingesetzt. Das beste der drei Verfahren (C4.5, [Quinlan 1993]) erbringt 67% Recall bei 71% Precision. Die Verbesserung des Ergebnisses gegenüber [Kupiec et al. 1995] dürfte (mit den oben erwähnten Einschränkungen bezüglich Vergleichbarkeit verschiedener Arbeiten) durch die verbesserten inhaltsbezogenen Attribute (Wortcharakteristik) erreicht worden sein. Eine themenabhängige Satzextraktion wurde bei dieser Arbeit auch durchgeführt und wird im nächsten Abschnitt erläutert.

Einen anderen Weg gehen Verfahren, die lexikalisches Wissen über Wörter und ihre Verwandtschaft (*Lexical Cohesion*) ausnutzen, um Sätze auszuwählen, die zur wichtigsten Gruppe von Wörtern gehören. Ein Beispiel ist die Verwendung von sogenannten *Lexical Chains* in [Barzilay und Elhadad 1999] und anderen Arbeiten. Eine Lexical Chain besteht aus Wörtern eines Textes, zwischen denen ausgewählte semantische Relationen bestehen. Die Wörter einer Kette verteilen sich typischerweise über mehrere Sätze, aber nur wenige Absätze. Ein Text kann mehrere solcher Ketten enthalten. Eine Kette repräsentiert ein Unterthema des Textes, das über einige Absätze behandelt wird; mit dem Wechsel der Ketten ist, so die Heuristik, ein Themawechsel verbunden. Ein Beispiel für einen Absatz mit zwei Ketten ist der folgende aus [Morris und Hirst 1991] (zitiert nach [Hovy und Marcu 1998]), wobei die Wörter der beiden Ketten durch verschiedene Schriftarten hervorgehoben sind:

> But Mr. Kenny's move speeded up work on a machine which uses micro-computers to control the rate at which an *anaesthetic* is pumped into the blood of *patients* undergoing *surgery*. Such machines are nothing new. But Mr. Kenny's device uses two personal computers to achieve much closer monitoring of the pump feeding the *anaesthetic* into the *patient*. Extensive testing of the equipment has sufficiently impressed the authorities which regulate *medical* equipment in Britain, and, so far, four other countries, to make this the first such machine to be licensed for commercial sale to *hospitals*.

Verschiedene Ketten können nach ihrer Länge und der Art ihrer semantischen Relationen gewichtet werden. Schließlich kann man diejenigen Sätze, in denen die meisten Wörter der stärksten Ketten auftauchen, für einen Extrakt verwenden. Damit konnten 67% Recall bei 61% Precision erzielt werden ([Barzilay und Elhadad 1999]). Quelle für die semantischen Relationen ist ein Thesaurus in elektronischer Form. Viele Arbeiten verwenden WordNet ([Miller 1995]), eine lexikalische Datenbank für die englische Sprache, in der semantisch verwandte Wortgruppen als sogenannte *Synsets*, Synonymmengen, organisiert sind. Eine solche Menge steht für ein semantisches Konzept, zu dem die enthaltenen Wörter gehören. Die Mengen sind durch die semantischen Relationen Synonymie, Antonymie (Gegensätzlichkeit), Meronymie (Teil-Ganzes-Beziehung) und Hyponymie („ist ein"-Beziehung) verbunden. Für die deutsche Sprache steht inzwischen auch GermaNet zur Verfügung ([Hamp und Feldweg 1997]), das in seinem Aufbau WordNet nachempfunden wurde.

Für gezieltes Satzfiltern wäre es möglich, eine textunabhängige Kette von Wörtern zum vorgegebenen Thema aufzubauen und damit direkt beliebige Sätze zu gewichten. Oder es könnten zu jedem Text seine Ketten gefunden werden und mit der vorgegebenen Kette verglichen werden; bei genügender Ähnlichkeit würden dann die Sätze, die die ähnlichste Kette berührt, extrahiert. Die im nächsten Kapitel beschriebenen Verfahren benutzen Listen von Wörtern zum gezielten Satzfiltern, die aber anders gewonnen werden. Es wäre interessant, die Versuche dieser Diplomarbeit zum Vergleich mit einer auf GermaNet basierenden Wortliste durchzuführen.

3.5.2. Gezielte Zusammenfassungen

Bei automatischer Textzusammenfassung ist die Evaluation mit Schwierigkeiten verbunden. Es stellt sich die Frage, wie man beispielsweise zwei verschiedene Zusammenfassungen desselben Textes vergleicht oder wie Systeme zu vergleichen sind, die unterschiedliche Längen der Zusammenfassungen, unterschiedliche Methoden, Schwerpunkte oder Textgenres verwenden. Zur einheitlichen Bewertung von Zusammenfassungen wurde 1998 ein MUC-vergleichbares Projekt namens SUMMAC durchgeführt, in dem verschiedene Zusammenfassungssysteme nach gleichen Kriterien verglichen wurden (siehe unten sowie [HAND und SUNDHEIM 1998]).

Eine nützliche Unterscheidung ist die zwischen intrinsischer und extrinsischer Bewertung ([SPARCK-JONES und GALLIERS 1996]). Intrinsische Bewertungen bewerten eine Zusammenfassung nach ihrer Qualität. Dazu gehören die Bewertungen der Satzextraktion (Recall, Precision und andere, Kapitel 4 und 6) wie auch Lesbarkeit oder Verständlichkeit. Extrinsische Bewertungen untersuchen die Nutzbarkeit der Zusammenfassung für bestimmte Aufgaben, beispielsweise die Zuordnung des Textes zu einer von mehreren vorgegebenen Klassen durch menschliche Testpersonen. Wenn die Zuordnung der Zusammenfassungen genauso akkurat möglich ist wie die der Ausgangstexte, aber schneller geht, weil die Testpersonen nicht den ganzen Text lesen müssen, so sind die Zusammenfassungen offensichtlich nutzbar. In [JING et al. 1998] werden drei Textzusammenfassungssysteme sowohl intrinsisch als auch extrinsisch bewertet, weniger um die Systeme zu testen, sondern um mehr über die Evaluation solcher Systeme zu lernen. Ein Resultat ist beispielsweise, dass die Länge der erstellten Zusammenfassung eine große Rolle spielt und ebenso die Textüberschrift, weshalb die Autoren vorschlagen, die Länge einer Zusammenfassung nicht starr vorzugeben sowie zur Bewertung nur der Zusammenfassungen die Textüberschriften den Testpersonen vorzuenthalten.

Eine Möglichkeit der intrinsischen Bewertung ist, den Testern Fragen zum Text zu stellen, obwohl sie nur die Zusammenfassung kennen. Dadurch lässt sich der *Informationsgehalt* der Zusammenfassungen messen. Hierbei stellt sich das Problem, dass bei der Erstellung generischer Zusammenfassungen nicht bekannt ist, für welchen Zweck sie dienen sollen. Dies gilt für menschlich wie maschinell erstellte Zusammenfassungen. Da sie möglichst allgemein den Textinhalt wiederspiegeln sollen, können sie spezifische Fragen oft nicht beantworten. Deshalb befassen sich einige Arbeiten mit Zusammenfassungen, die auf ein bestimmtes Thema oder eine Benutzeranfrage abzielen und andere Inhalte der Texte ignorieren. Die englischen Bezeichnungen dafür variieren (*query-oriented summary*, [HOVY und MARCU 1998]; *tailored summary*, [HAHN und MANI 1998]; *domain-specific summary*, [RILOFF 1993]). Ich werde die oben schon eingeführte Bezeichnung *gezielte Zusammenfassung* verwenden, analog zum gezielten Satzfiltern. Wie schon erwähnt wurde, erinnert diese Art der Zusammenfassung an Informationsextraktion im weiteren Sinne. Charakteristisch hierfür ist, den Lesern den Satzzusammenhang, in dem die Information formuliert ist, zu erhalten. Daraus kann sehr oft weiterer Nutzen gezogen werden und Missverständnissen vorgebeugt werden. Letztendlich läuft dieses Vorgehen darauf hinaus, das genaue Textverständnis den Lesern zu überlassen und auch, im Gegensatz zur Informationsextraktion im Sinne von Abschnitt 3.4, zu ermöglichen.

Bei gezielten Zusammenfassungen ist es, da ihr Zweck bekannt ist, leichter, genaue Fragen zu stellen und so den Informationsgehalt besser anzunähern. Insbesondere lassen sich verschiedene Systeme, die die gleiche inhaltliche Vorgabe für gezielte Zusammenfassungen bekommen, mit einem festgelegten Katalog von Fragen an Leser der Zusammenfassungen vergleichen. Abschnitt 6.3.2 erläutert die Bewertung durch gezielte Fragen an Testpersonen im Rahmen dieser Arbeit. Im Rahmen von SUMMAC wurde eine ähnliche Bewertung durchgeführt, ebenfalls für gezielte Zusammenfassungen, indem nicht Testpersonen befragt wurden, sondern die An- oder Abwesenheit von Textpassagen, die für die Antwort auf eine gegebene Frage relevant sind, gemessen wurde.

Die Spezifikation der gesuchten Information kann auf verschiedene Weisen geschehen. Wie in Abschnitt 3.4 erläutert wurde, gibt man bei der klassischen Informationsextraktion genaue Schablonen vor. In [RILOFF 1993] findet sich der Vorschlag, die gezielte Zusammenfassung auf den Ergebnissen eines Informationsextraktionssystems aufzubauen; die vorgeschlagene Methode wurde jedoch nicht getestet. Eine andere Möglichkeit der inhaltlichen Vorgabe ist eine Liste von relevanten Wörtern. Darauf beruht das gezielte Satzfiltern, das im nächsten Kapitel vorgestellt wird; die Wortliste wird dabei automatisch gewonnen. Für das SUMMAC-Projekt war die inhaltliche Vorgabe eine Benutzeranfrage, wie sie für Information Retrieval üblich ist. Aus den Veröffentlichungen zu SUMMAC ist jedoch nicht zu entnehmen, zu welchem Grad die einzelnen teilnehmenden Systeme die Benutzeranfragen einfließen ließen.

Im folgenden werden einige Arbeiten vorgestellt, die sich mit gezielten Zusammenfassungen beschäftigt haben. Es wird deutlich, dass hier bereits gezieltes Satzfiltern verwendet wird, da auch diese Arbeiten Extrakte erstellen. Anregungen aus diesen Arbeiten werden in Kapitel 4 aufgenommen, um den eigenen Ansatz zu entwickeln.

In der auf Seite 24 erwähnten Arbeit [MANI und BLOEDORN 1998] dienen schon erstellte Zusammenfassungen als Vorgabe für maschinelle Lernverfahren. Die dabei verwendeten Zusammenfassungen sind generisch. Die Autoren wollten jedoch auch gezielte Zusammenfassung ermöglichen und ließen dazu gezielte Extrakte automatisch erstellen, um aus ihnen zu lernen. Diese Extrakte wurden mit Hilfe einer Wortliste gebildet, die wiederum auf vorher für themenrelevant befundenen Texten beruht[5]. Die gezielten Extrakte (mit fester Länge) wurden für alle Texte des Korpus gebildet, auch wenn sie nicht zum vorgegebenen Thema passten. Mit diesen Extrakten wurde dann gelernt wie oben (auf Seite 24) geschildert. Die Ergebniswerte der Satzklassifikation liegen deutlich über jenen bei der generischen Zusammenfassung und erreichen 91% Recall bei 88% Precision. Die Bedeutung dieses Ergebnisses muss jedoch relativiert werden: Die Anzahl der Themenwörter, die der Extrakterstellung zugrunde lag, wurde auch als Attribut beim Lernen verwendet. Damit wird die Lernaufgabe deutlich erleichtert und die Ergebnisse sollten in diesem Licht gesehen werden. Es ist unklar, ob dieses Verfahren mit von Menschen vorgegebenen Trainingsbeispielen ähnlich gut funktionieren würde, zumal dann nicht jeder Text eine Zusammenfassung erhielte, um aus ihr zu lernen, sondern nur die themenrelevanten Texte.

Gezielte Zusammenfassungen verwenden auch [TOMBROS und SANDERSON 1998] im Umfeld von Information Retrieval. Ihre Zusammenfassungen werden im Rahmen eines IR-Systems den Benutzern präsentiert, damit sie die Relevanz der Texte einschätzen können, die das IR-System auf die Benutzeranfrage hin gefunden hat. Neben einigen

[5]Die Ermittlung der Themenwörter erfolgte mit der G^2-Statistik, die im folgenden Kapitel, Abschnitt 4.2.3, erläutert wird.

der schon erwähnten Methoden zur Satzgewichtung setzen die Autoren das Gewicht der Sätze hoch, wenn in ihnen Wörter aus der Benutzeranfrage vorkommen. Genauer: Wenn ein Satz n Wörter aus der Benutzeranfrage enthält und die Benutzeranfrage selbst m Wörter, so wird das Gewicht des Satzes um n^2/m erhöht. Die Extrakte tendieren also dazu, mehr themenrelevante Sätze zu enthalten. Dazu wurde eine extrinsische Bewertung im Rahmen des eingesetzten IR-Systems durchgeführt. Zwei Benutzergruppen beurteilten die Relevanz von Dokumenten, die das System auf vorgegebene Anfragen hin präsentierte. Die eine Gruppe hatte dabei Zugriff auf die gezielten Zusammenfassungen, während die andere nur die ersten Zeilen eines Textes sah. Beide Gruppen konnten per Hyperlink jedes Dokument auch ganz ansehen. Die Benutzer mit Zugriff auf die Zusammenfassungen konnten die Relevanz der Dokumente schneller, besser und mit weniger Zugriffen auf das ganze Dokument beurteilen, als die andere Gruppe. Allerdings geht daraus nicht hervor, welchen Einfluss die Zielgerichtetheit der Zusammenfassungen hatte; ein Vergleich mit generischen Extrakten wäre wünschenswert.

Ein weiterer Ansatz zu gezielter Zusammenfassung, der auf dem Lexical Cohesion-Prinzip beruht, findet sich in [Baldwin und Morton 1998]. Darin werden „Assoziationen" zwischen Wörtern in der Benutzeranfrage und im Text verwendet, um Sätze auszuwählen, die stark mit der Anfrage assoziiert werden. Eine Assoziation zwischen einem Wort im Text und einem in der Anfrage beruht auf der Wahrscheinlichkeit, dass die beiden Wörter generell zusammen, das heißt im gleichen Text, auftreten; diese Wahrscheinlichkeiten wurden durch Auszählung in einem großen Korpus abgeschätzt. Assoziationen bestehen also vornehmlich zwischen inhaltlich verwandten Wörtern. Weitere Assoziationen entstehen durch Vergleich der Zeichenketten (bei Nomen) sowie durch spezielle Wörterbücher, die Städtenamen und Staaten in Verbindung setzen, so dass auf eine Anfrage nach Ereignissen in Deutschland Berichte über Dortmund in Betracht gezogen werden können. Zur Bewertung der Zusammenfassungen dienten wieder Testpersonen, die die Relevanz von Texten für die Anfrage aufgrund der Zusammenfassungen beurteilen sollten, also eine extrinsische Bewertung, die gute Ergebnisse brachte. Auch hier fehlt aber der Vergleich mit nicht anfrageorientierten Zusammenfassungen.

Zusammenfassend lässt sich sagen, dass zur gezielten Zusammenfassung bisher wenig Literatur existiert, auch wenn die obigen Beispiele keinen Anspruch auf Vollständigkeit erheben. Viele der Arbeiten zu diesem Thema beschäftigen sich mit der Aufgabe, die Relevanz eines Dokuments anhand der Zusammenfassung besser beurteilen zu können, unterstützen also Information Retrieval. Die Aufgabenstellung der Summac-Konferenz 1998 (siehe oben) mag hier eine Rolle gespielt haben (vgl. [Sanderson 1998]). Die Arbeiten verwenden also eine Benutzeranfrage zur Vorgabe des Themas, was nicht zur Aufgabe dieser Diplomarbeit passt. Auch ist die Beurteilung der Zielgerichtetheit (als Verbesserung für bestimmte Aufgaben gegenüber generischen Extrakten) bisher zu kurz gekommen. Indem der Informationsgehalt der gezielten Zusammenfassungen durch Testpersonen überprüft wird, untersucht diese Diplomarbeit diesen Aspekt genauer.

3.6. Zusammenfassung des Kapitels

In diesem Kapitel wurden die wesentlichen Text Mining-Methoden vorgestellt. Methoden zur Suche in Texten nach für diesen Text wichtigen Sätzen sowie fokussiert nach Sätzen, die zu einem vorgegebenen Thema gehören, wurden diskutiert (Satzfiltern bzw. gezieltes Satzfiltern). Die letzteren Methoden wurden zur gezielten Textzusammenfassung zugeordnet, im weiteren Sinne lassen sie sich zur Informationsextraktion rechnen. Eine andere Sichtweise ist die der satzweisen Klassifikation von Sätzen als zum Thema gehörend oder nicht, weshalb Methoden zur Textklassifikation ebenfalls vorgestellt wurden. Allerdings stellt die Kürze von Sätzen im Vergleich zu ganzen Texten eine Schwierigkeit für Textklassifikationsmethoden dar.

Die vorgestellten Methoden zum Satzfiltern lassen sich nur bedingt anwenden auf gezieltes Satzfiltern; entsprechende Hinweise wurden für jede Methode gegeben. Die Methoden zur gezielten Zusammenfassung verlassen sich auf das Vorhandensein einer Benutzeranfrage oder lernen aus vorgegebenen Zusammenfassungen. Für die Aufgabenstellung dieser Diplomarbeit kann aber beides nicht vorausgesetzt werden. Stattdessen wird eine inhaltliche Vorgabe automatisch aus einem Korpus ermittelt. Dies geschieht in Form von Wortlisten, deren automatische Ermittlung und Verwendung zum gezielten Satzfiltern das nächste Kapitel beschreibt.

4. Gezieltes Satzfiltern

Im vorigen Kapitel wurden einige Methoden der themengerichteten (gezielten) Extrakterstellung vorgestellt, darunter einige, die maschinelles Lernen einsetzen. Die Vorgabe des gewünschten Themas geschieht dabei in der Regel durch Benutzeranfragen, alternativ durch schon erstellte Zusammenfassungen von Texten, die zum Thema gehören. Dieses Kapitel stellt eine Methode vor, die auf beides verzichten kann und nur auf der Markierung einer Beispielmenge von Texten beruht. Sie wurde im Rahmen dieser Diplomarbeit implementiert und getestet; die Ergebnisse beschreibt Kapitel 6.

4.1. Grundidee

Aus einer Beispielmenge von Texten wird automatisch eine Stichwortliste von Wörtern, die zum gewünschten Thema gehören, gewonnen. Damit befasst sich Abschnitt 4.2. Die Stichwortgewinnung geschieht „offline", das heißt sie wird einmalig durchgeführt, danach dient die fertige Liste zum Satzfiltern. Die Themenwörter erhalten dabei jeweils ein Gewicht, das angibt, wie charakteristisch das Wort für das Thema ist. Man kann das Wortgewicht als angenäherte Wahrscheinlichkeit interpretieren, mit der ein Satz, in dem das Wort vorkommt, zum Thema gehört. Die Wortgewichte eines Satzes addieren sich zum Satzgewicht. Alle Sätze, deren Gewicht über einem zu bestimmenden Schwellwert liegt, werden als zum Thema gehörend betrachtet. Sätze, in denen kein Wort aus der Stichwortliste vorkommt, haben Gewicht 0 und gehören nicht zum Thema. Die themenspezifischen Extrakte bestehen aus allen Sätzen mit genügendem Gewicht. Die Satzauswahl über den Schwellwert wird in Abschnitt 4.3 behandelt.

Um die Stichwortliste automatisch gewinnen zu können, muss die Beispielmenge von Texten—im folgenden Trainingsmenge genannt—satzweise oder textweise markiert werden. Die Markierung sollte durch Personen erfolgen, die das Thema hinreichend genau eingrenzen können, also beispielsweise durch die späteren Benutzer der gefilterten Extrakte. Textweise Markierung bedeutet, dass jeder Text, der mindestens eine Passage enthält, die zum Thema gehört, als relevant markiert wird. Bei satzweiser Markierung erhält jeder Satz, der zum Thema gehört, eine Markierung. Letzteres Vorgehen ist also mit deutlich höherem Aufwand verbunden, bietet dafür aber eine bessere Genauigkeit beim Auffinden der Stichwörter, denn bei der textweisen Markierung werden markierte Texte auch Passagen enthalten, die nicht zum Thema gehören (falls nicht alle markierten Texte so beschaffen sind, dass sie ausschließlich das Thema behandeln). Für diese Diplomarbeit wurde eine satzweise Markierung vorgenommen, die natürlich eine textweise Markierung direkt impliziert (entweder kommen in einem Text markierte Sätze vor oder nicht). Die Auswirkungen der verschiedenen Markierungsweisen wurden getestet (Abschnitt 6.2 nennt die Resultate).

Dieses Verfahren zum Satzfiltern kombiniert einige der Vorteile der Anwendung maschinellen Lernens mit weiteren positiven Aspekten:

- Die Vorgabe des gewünschten Themas muss nicht explizit erfolgen. Es ist also nicht notwendig, Regeln festzulegen, die angeben, wann ein Satz zum Thema gehört. Statt dessen erschließt sich das Thema implizit aus den Markierungen.
- Die Markierung von Texten oder Sätzen kann leicht auch durch ungeschulte Benutzer vorgenommen werden; kein Spezialwissen ist dazu erforderlich.
- Das eigentliche Satzfiltern benötigt nur eine Stichwortliste mit Gewichten und ist sehr effizient. Für die Wortgewichte kann eine effiziente Dictionary-Datenstruktur verwendet werden. Da nur ein kleiner Teil der Wörter eines Textes in der Rangliste vorkommt, bestimmen viele erfolglose Lookups die Laufzeit. Die Verwendung dynamischer offener Hashingtabellen bietet sich an, bei denen die Rechenzeit der Lookups von der Auslastung der Tabelle abhängt. Bei den eher kleinen Wortanzahlen, die hier vorkommen, kann diese problemlos klein gehalten werden, womit die Zeit für n Lookups durch $O(n)$ gut angenähert werden kann.
- Über die Veränderung des Schwellwertes (des Mindestgewichtes für einen Satz, um zum Extrakt zu gehören) lässt sich die Satzauswahl leicht beeinflussen; siehe Abschnitt 4.3.
- Die Verfahren zur Gewinnung der Stichwortliste (siehe folgenden Abschnitt) behandeln die Fälle der satzweisen und der textweisen Markierung gleich, so dass beides je nach möglichem Aufwand verwendet werden kann.
- Wenn das gewünschte Thema sich nicht mit der Zeit ändert, muss die Stichwortliste nur einmal ermittelt werden.
- Die Verwendung mehrerer Stichwortlisten (zu verschiedenen Themen) ist leicht möglich.
- Zur Änderung eines Themas oder Hinzunahme eines weiteren Themas muss lediglich eine Markierung von Trainingstexten erfolgen, mit der dann einmalig eine Stichwortliste gewonnen werden kann.
- Das Verfahren an sich ist sprachunabhängig und setzt lediglich die Vorverarbeitung von Texten bis zur Stammformenreduktion, die natürlich sprachabhängig ist, voraus (vgl. folgenden Abschnitt).
- Die Berechnung der Stichwortliste stützt sich auf markierte Textteile, bezieht also den Kontext von Wörtern mit ein: beispielsweise wird ein Wort mit zwei Bedeutungen, von denen nur eine für das vorgegebene Thema relevant ist, im Allgemeinen kein hohes Gewicht erhalten, da es sowohl in markierten Textteilen (mit der ersten Bedeutung) als auch in nicht markierten (mit der zweiten) vorkommen kann. Die Probleme der Polysemie und Synonymie werden also abgemildert. Dies schließt nicht aus, dass sie in Einzelfällen für falsche Ergebnisse sorgen.

- Die aufwändige Erstellung fertiger Zusammenfassungen, um aus ihnen zu lernen, ist nicht notwendig.
- Die inhaltliche Vorgabe ist präziser als bei Benutzeranfragen, die typischerweise recht kurz sind.

Als Nachteil des Verfahrens ist der nicht unbeträchtliche Aufwand zur Markierung einzelner Sätze zu nennen. Die textweise Markierung ist zwar deutlich weniger aufwändig, aber auch weniger genau und führt zu schlechteren Resultaten (vgl. Abschnitt 6.2). Auch verzögert die Stammformenreduktion die Filterung von Texten merklich, was bei großen Textmengen zu mangelnder Performanz führen kann. Will man also beispielsweise Emails filtern, so empfiehlt sich statt einer zentralen Filterung auf einem Mailserver die dezentrale Verteilung auf einzelne Clients. Allerdings setzen auch viele andere Text Mining-Methoden Stammformenreduktion ein.

4.2. Gewinnung der Stichwortliste

Es gibt eine Anzahl von Arbeiten, die sich mit automatischer Extraktion von Stichwörtern aus Texten befasst haben, ob zur inhaltlichen Charakterisierung des Textes in Kurzform, zur Automatisierung der Erstellung eines Indexes oder zu anderen Zwecken. Den Arbeiten ist jedoch gemein, dass sie versuchen, alle Wörter zu finden, die den Inhalt eines Textes charakterisieren. Alle wesentlichen inhaltlichen Aspekte des Textes sollen also abgedeckt werden. Dagegen geht es bei den Verfahren für diese Diplomarbeit darum, nur Wörter zu einem vorgegebenen Thema zu finden. Die Wichtigkeit eines Wortes muss also in Bezug auf ein Thema errechnet werden, nicht in Bezug auf seinen Text.

Beispielsweise wird in [Turney 2000] maschinelles Lernen zur Erkennung von Schlagwörtern verwendet. Darin findet sich auch eine Übersicht über verschiedene Stichworterkennungsverfahren, an die sich einige der folgenden kurzen Darstellungen anlehnen.

Schlagwörter deuten für menschliche Leser eine Kategorie für den Text an; es werden typischerweise fünf bis fünfzehn Schlagwörter pro Text identifiziert. In der erwähnten Arbeit werden im wesentlichen die Häufigkeit eines Wortes oder einer mehrwortigen Phrase sowie die Position des ersten Auftauchens im Text, relativ zur Länge des Textes, als Attribute verwendet, um das Wort oder die Phrase als Schlagwort zu klassifizieren oder nicht. Als Lernverfahren werden *C4.5* ([Quinlan 1993]) und ein genetischer Algorithmus eingesetzt. Während die höchsten Precision-Werte der Resultate (zu Precision siehe Abschnitt 4.3) im Vergleich zu den vorgegebenen Schlagwörtern unter 30% liegen, werden gut 60% der extrahierten Schlagwörter von menschlichen Benutzern als aussagekräftig genug empfunden.

Darauf aufbauend beschreiben [Frank et al. 1999] eine in gewisser Weise themenorientierte Schlagwortextraktion. Zunächst wird durch Verwendung von *Naive Bayes Classification* als Lernverfahren die Trainingszeit stark gesenkt, ohne signifikante Einbußen bei den Ergebnissen. Als weiteres Resultat wird erläutert, dass die Schlagwortextraktion am besten funktioniert, wenn das Training auf Texten aus der passenden inhaltlichen Domäne erfolgte. Auf diesem Ergebnis aufbauend, nehmen die Autoren ein zusätzliches Lernattribut hinzu, das abhängig von der Domäne ist, nämlich die Häufigkeit des Vorkommens eines Schlagwortkandidaten in den Trainingstexten. Die Angabe des Themas erfolgt also implizit durch die Sammlung der Texte aus einer Domäne; würden nach dem Training Texte einer anderen Domäne verwendet, so würde die Angabe der Häufigkeit eines Kandidaten in den Trainingstexten wenig Sinn machen. So werden also die Ergebnisse für die jeweilige Domäne verbessert.

Im Vergleich zu diesen Verfahren muss sich eine Stichwortliste für diese Diplomarbeit nicht auf wenige, für Menschen aussagekräftige Wörter beschränken; sie kann vielmehr beliebig lang sein und beliebige Wortarten enthalten. Dementsprechend einfacher ist die Gewinnung.

Auf der anderen Seite des Spektrums, im Sinne der Anzahl zu extrahierender Stichwörter, steht die Indizierung eines Textes für Suchmaschinen, die nahezu alle Wörter eines Textes verwendet. In der Mitte finden sich Verfahren zur automatischen Erstellung von Indexen zum Nachschlagen, wie sie am Ende von Büchern üblich sind. Zum Beispiel beschreibt [Nakagawa 1997] die Extraktion von Nomen zu diesem Zweck, die auf der Häufigkeit des Auftauchens im Text basiert. Einen anderen Weg geht die Arbeit [Leung und Kan 1997], indem Wörter von einer vorgegebenen Liste möglicher Indexwörter möglichen Texten zugeordnet werden. Die Zuordnung geschieht mit maschinellem Lernen, indem, vereinfacht gesagt, die Häufigkeit eines Wortes von der Liste in für dieses Wort relevanten Texten mit der in nichtrelevanten Texten verglichen wird. Hier stellt sich das Problem, die Liste der möglichen Stichwörter vorgeben zu müssen, während sie für diese Diplomarbeit ja erst gewonnen werden soll. Themenorientierte Ansätze sind mir auf diesem Gebiet nicht bekannt und machen für Nachschlaglisten auch wenig Sinn.

In [Cohen 1995] werden domänenunabhängig Indexwörter zum Zweck der Übersicht über einen Textinhalt gewonnen, ähnlich der Schlagwortextraktion, aber ohne die strenge Begrenzung der Anzahl. Das Verfahren beruht auf n-Grammen (siehe Abschnitt 3.2), was den Vorteil der Sprachunabhängigkeit hat. Zur Gewichtung eines n-Gramms, die zur Gewichtung eines Wortes führt, dient die G^2-Statistik, die unten in 4.2.3 vorgestellt wird und dort zur themenorientierten Gewichtung herangezogen wird.

Im folgenden (Abschnitte 4.2.1 bis 4.2.5) werden fünf verschiedene Verfahren zur automatischen Gewinnung einer Rangliste von themenspezifischen Wörtern aus markierten Texten beschrieben, die in dieser Diplomarbeit implementiert und getestet wurden. Jedes Verfahren verwendet eine andere Methode, das Gewicht eines Wortes im Hinblick auf das vorgegebene Thema zu berechnen. Die Rangliste ist die nach dem Gewicht absteigend sortierte Liste aller Wörter mit positivem Gewicht. Es wurde schon darauf hingewiesen, dass die textweise und die satzweise Markierung gleich behandelt werden können. In diesem Abschnitt werde ich daher das Wort „Text“ stellvertretend für „Text“ und „Satz“ verwenden.

Um die Gewinnung der Stichwortliste zu erleichtern, kann eine Stammformenreduktion der Trainingstexte durchgeführt werden. Dadurch werden verschiedene Formen desselben Wortes nicht als unterschiedliche Stichwörter mit unterschiedlichen Gewichten aufgefasst. Dies erzwingt dann auch eine Stammformenreduktion neuer, zu filternder Texte, da nur Stammformen in der Liste stehen. Die Auswertung in Kapitel 6 wird zeigen, dass in dieser Diplomarbeit ohne Stammformenreduktion keine guten Ergebnisse erzielt wurden.

Weiterhin kann man der Stichwortgewinnung auch die Eliminierung von Stoppworten vorausgehen lassen (Abschnitt 3.2). Vorsicht ist jedoch angebracht, da man schlecht vorhersehen kann, ob bestimmte Funktionswörter nicht doch zur Unterscheidung von Themen dienen können. Bei der Termindomäne stellen sich beispielsweise die Präpositionen `am` und `um` als charakteristisch für die Domäne heraus, da sie viele Zeitangaben wie `am Samstag um 11 Uhr` enthält. Daher werden für die unten beschriebene Stichwortgewinnung nur Artikel sowie Konjunktionen wie `und`, `aber` oder `dass` aus den Texten entfernt.

4.2.1. Worthäufigkeit

Das einfachste Verfahren besteht darin, die Häufigkeit jedes Wortes in positiv markierten Texten mit der in den anderen Texten in Beziehung zu setzen. Ist p die Anzahl der Vorkommen eines Wortes in den positiv markierten Texten und n die Anzahl in den anderen, so wird das Wortgewicht zu p/n berechnet. Der Ansatz lässt sich verbessern, indem die relative Häufigkeit eines Wortes in den positiven Texten durch die in den anderen geteilt wird. Mit p und n wie zuvor sowie s_p als Anzahl der positiven Texte und s_n als Anzahl der negativen berechnet sich das Wortgewicht zu

$$\frac{p/s_p}{n/s_n}\,. \tag{4.1}$$

Kommt das Wort nur in den positiven Texten vor ($n = 0$), so ergibt p/s_p allein das Gewicht. Die Verbesserung besteht darin, dass der so erhaltene Wert unabhängig von der jeweiligen Häufigkeit von positiven und negativen Texten im Korpus ist. Für die Experimente dieser Arbeit wurde diese zweite Berechnungsmethode gewählt. Zu beachten ist, dass keine Wörter, die nur in negativen Texten vorkommen, ein Gewicht ungleich 0 erhalten können. Die Laufzeit für dieses Verfahren ist beschränkt durch das Produkt aus der Anzahl der Wörter in der Textsammlung und der Anzahl der *verschiedenen* Wörter in der Textsammlung; durch die Verwendung effizienter Datenstrukturen wie Hashingtabellen zur Verwaltung der Worthäufigkeiten kann die Laufzeit gesenkt werden.

4.2.2. Tf-Idf-Gewichtung

In Abschnitt 3.2 wird das Tf-Idf-Maß vorgestellt, das für einen Text angibt, wie charakteristisch ein bestimmtes Wort für ihn ist. Um ein Gewicht jedes Wortes unabhängig vom Text zu erhalten, muss man alle Texte betrachten, in denen es vorkommt. Zur themenspezifischen Gewichtung wird die Summe p seiner Tf-Idf-Werte über alle positiven Texte berechnet sowie die Summe n seiner Tf-Idf-Werte über alle negativen Texte. Mit s_p und s_n wie oben errechnet sich dann das Wortgewicht mit derselben Formel (4.1) wie bei der Worthäufigkeit. Die Laufzeit ist ebenfalls dieselbe.

4.2.3. G^2-Statistik

In der auf Seite 27 erwähnten Arbeit [Mani und Bloedorn 1998] wurde erwähnt, dass die Autoren ein statistisches Maß namens G^2 verwenden, um die wichtigsten Wörter einer Sammlung von Texten zu finden. Ihre Version dieses Maßes beruht auf [Cohen 1995]. Man kann damit annähernd ermitteln, ob die Häufigkeit eines Wortes in seinem Text größer ist, als aufgrund seiner Häufigkeit im Gesamtkorpus zu erwarten wäre, unter Berücksichtigung des Größenverhältnisses zwischen dem Text und dem Korpus.

Die Formel, die dazu in [COHEN 1995] hergeleitet wird, basiert auf der Annahme, dass die beobachteten Worthäufigkeiten eine Realisierung der zugrundeliegenden, unbekannten Wahrscheinlichkeiten des Auftretens der Wörter in den betrachteten Texten sind. Genauer gesagt, ist $\vec{c_o} = (c_1, c_2, \ldots, c_m)$ der Vektor, der die Worthäufigkeiten der m Wörter eines Textes d enthält, so ist die Wahrscheinlichkeit des Auftretens von $\vec{c_o}$ abhängig vom unbekannten Vektor $\vec{p} = (p_1, p_2, \ldots, p_m)$ sowie von S, der Anzahl aller Wörter des Textes, wobei $\vec{p}$ die Wahrscheinlichkeiten angibt, mit denen die m Wörter auftreten:

$$Pr(\vec{c_o} = \vec{c}) = f(\vec{c}\,|\,\vec{p}, S)$$

Die Funktion f ist in [COHEN 1995] angegeben. Bezeichne analog $\vec{q}$ die unbekannten Wahrscheinlichkeiten des Auftretens der Wörter im Gesamtkorpus, für den die Worthäufigkeiten $\vec{b}$ beobachtet werden, und sei die Anzahl der Wörter im Gesamtkorpus R. Dann gilt es, zur Bewertung der Relevanz eines Wortes t_i für den Text d die Wahrscheinlichkeit zu schätzen, dass $p_i > q_i$ gilt bei den gegebenen Beobachtungen c_i und b_i, das heißt dass die Wahrscheinlichkeit des Auftretens von t_i im Text höher ist als die des Auftretens im Hintergrundkorpus. Der G^2-Wert $w(t_i)$ leistet genau dies und berechnet sich zu

$$w(t_i) = \begin{cases} 0 & \text{falls} \quad c_i/S < b_i/R \\ c_i \ln(c_i/S) + b_i \ln(b_i/R) - (c_i + b_i) \ln\left(\frac{c_i + b_i}{S + R}\right) & \text{sonst} \end{cases} \tag{4.2}$$

Der obere Wert ist 0, weil in dem Fall, dass die Häufigkeit von t_i im Text nach unten abweicht im Vergleich zum Gesamtkorpus, zwar eine statistische Abweichung vorliegt, aber im Blick auf das Auffinden wichtiger Wörter dieser Fall sinnvollerweise ignoriert werden sollte.

Diese Formel kann auch zur themenbezogenen Wortgewichtung verwendet werden, indem nicht ein Text und der Gesamtkorpus verglichen werden, sondern alle positiv markierten Texte mit allen anderen (nicht markierten). Es wird also geschätzt, ob das Wort in positiv markierten Texten wahrscheinlicher ist als in allgemeinen, bei den gegebenen Worthäufigkeiten. Dabei wird angenommen, dass die nicht markierten Texte repräsentativ für beliebige Texte sind. Dies ist nicht ganz richtig, denn auch markierte Texte gehören zum allgemeinen Textaufkommen. Für die Experimente in dieser Diplomarbeit sind jedoch die markierten Texte in den verwendeten Korpora überrepräsentiert (Abschnitt 6.1), so dass sie nicht alle in den Hintergrundkorpus einbezogen werden sollten, sondern nur anteilig entsprechend ihrer tatsächlichen Häufigkeit. Diese Häufigkeit ist jedoch unbekannt, weshalb darauf in dieser Arbeit verzichtet wurde. Zur Berechnung des G^2-Wertes des Wortes t_i wird Formel (4.2) verwendet, wobei den Variablen demnach folgende Bedeutungen zukommen:

- c_i ist die Anzahl der Vorkommen von t_i in positiv markierten Texten;
- b_i ist die Anzahl der Vorkommen von t_i in den anderen Texten;
- S ist die Anzahl aller Wörter in positiv markierten Texten;
- R ist die Anzahl aller Wörter in den anderen Texten.

Zu beachten ist, dass b_i jetzt den Wert 0 annehmen kann und der zweite Summand im unteren Teil von Formel (4.2) dann nicht mehr definiert ist. Für diesen Fall wird in dieser Diplomarbeit das G^2-Gewicht zu c_i/S bestimmt. Der G^2-Wert wird als Wortgewicht für die Rangliste übernommen. Wiederum ist die Laufzeit wie oben.

In den bisherigen Verfahren werden nur Wörter gewichtet, die in den positiv markierten Texten auftreten, während bei den folgenden Verfahren alle Wörter gewichtet werden.

4.2.4. Information Gain

Sowohl für die Worthäufigkeit als auch die Tf-Idf-Gewichtung wurde für diese Diplomarbeit das System `TCat` verwendet, das an der Carnegie Mellon University entwickelt wurde und das, neben Werkzeugen zur Textklassifikation, die Zählung der Häufigkeiten jedes Wortes in jedem Text übernimmt. Außerdem berechnet es für jedes Wort dessen *Information Gain*-Wert. Das Information Gain-Kriterium ist beispielsweise in [Yang und Pedersen 1997] beschrieben. Es misst die Anzahl der Informationsbits, die durch die An- oder Abwesenheit eines Wortes in einem bestimmten Text zur Vorhersage der Klasse des Textes gewonnen werden. Es beruht auf Abschätzungen der Wahrscheinlichkeiten $Pr(C_i|t)$, mit der ein Text die Klasse C_i hat, wenn das Wort t darin auftritt, und $Pr(C_i|\bar{t})$, mit der er diese Klasse hat, wenn das Wort nicht darin auftritt. Dazu kommen die a priori-Wahrscheinlichkeiten $Pr(C_i)$, mit der die Klasse C_i auftritt, $Pr(t)$, mit der das Wort t auftritt, und $Pr(\bar{t})$, mit der es nicht auftritt. Die Abschätzungen dieser Wahrscheinlichkeiten werden durch Abzählen aus den Trainingstexten gewonnen. Sei $\mathcal{C} = \{C_1, C_2, \ldots\}$ die Menge der möglichen Klassen. Der Information Gain-Wert $g(t)$ des Wortes t berechnet sich zu

$$\begin{aligned} g(t) = \quad & -\sum_{C\in\mathcal{C}} Pr(C) \log Pr(C) \\ & + Pr(t) \sum_{C\in\mathcal{C}} Pr(C \mid t) \log Pr(C \mid t) \\ & + Pr(\bar{t}) \sum_{C\in\mathcal{C}} Pr(C \mid \bar{t}) \log Pr(C \mid \bar{t}) \end{aligned}$$

Er wird als Wortgewicht für die Rangliste übernommen. Die Laufzeit entspricht der der vorigen Verfahren, multipliziert mit $|\mathcal{C}|$, der Anzahl der Klassen (wobei in dieser Arbeit nur binäre Klassifikationen betrachtet werden).

4.2.5. SVM-Gewichtung

In Abschnitt 3.3.1 werden Support Vector Machines (SVMs) als Klassifikationsverfahren vorgestellt. Wie dort erläutert wird, finden SVMs eine Hyperebene, die die Trainingsmenge möglichst fehlerfrei in positiv und negativ trennt und dabei den Abstand zu den nächstgelegenen Trainingspunkten maximiert. Die Gleichung der Hyperebene ist

$$\vec{w} \cdot \vec{x} + b = 0$$

mit Koeffizientenvektor $\vec{w}$ und Verschiebung vom Ursprung b. Da $\vec{x}$ eine wortbasierte Vektorrepräsentation des Textes ist, lässt sich jede Stelle von $\vec{w}$ interpretieren als Gewicht, mit dem das entsprechende Wort die Position der Hyperebene beeinflusst hat. Das Gewicht jedes Wortes wird also bei diesem Verfahren direkt aus $\vec{w}$ abgelesen, nachdem zuvor die SVM auf der gegebenen Trainingsmenge trainiert wurde.

Eine Variante dazu ist, dies zu kombinieren mit den Tf-Idf-Gewichten, indem nur das SVM-Gewicht von Wörtern berücksichtigt wird, die auch einen gewissen Mindest-Tf-Idf-Wert haben. Dadurch wird die Rangliste kürzer und führt auch, wie Kapitel 6 zeigt, zu besseren Resultaten. Hierbei wird also zunächst für jedes Wort die Summe seiner Tf-Idf-Werte über alle Texte bestimmt. Diese Gewichte werden anschließend auf den Bereich 0 bis 1 skaliert. Nur die Wörter, deren Gewichte dann über dem willkürlich festgelegten Schwellwert 0.1 liegen, werden in die Rangliste aufgenommen, und zwar sortiert nach dem von der SVM vergebenen Gewicht. Bei dieser Variante erhalten die Wörter demnach zunächst ein *nicht* themenspezifisches Tf-Idf-Gewicht, anschließend erhalten die Wörter, deren Gewicht hoch genug ist, ein themenspezifisches Gewicht durch die SVM.

Eine dritte Variante ist, die SVM nur auf Wörtern mit hohem Tf-Idf-Gewicht zu trainieren. Dies nennt man Attributselektion. Zwar wurden die Klassifikationsergebnisse der SVM damit in [Joachims 1997] nicht verbessert, aber es geht hier nur um die von der SVM vergebenen Wortgewichte. Einige Vorausexperimente für diese Arbeit zeigten aber auch keine Verbesserung der Ranglisten gegenüber der mit allen Wörtern trainierten SVM, so dass diese Variante nicht weiter verfolgt wurde.

Die Resultate der ersten beiden Varianten erläutert Kapitel 6. Die Laufzeit wird bestimmt durch die Trainingszeit der SVM, wofür effiziente Algorithmen existieren (vgl. Abschnitt 3.3.1).

4.3. Satzauswahl

Jedes Verfahren aus dem vorigen Abschnitt liefert eine nach Gewicht sortierte Rangliste von Stichwörtern. Zu den oben genannten Laufzeiten kommt daher noch die Rechenzeit für die Sortierung der Liste hinzu, bei n Wörtern also $O(n \log n)$. Zur besseren Vergleichbarkeit verschiedener Listen werden alle Gewichte auf den Bereich $[0 \ldots 1]$ skaliert, so dass das erste Wort der Liste das Gewicht 1 und das letzte das Gewicht 0 haben. Für die Satzextraktion ist es unerheblich, ob die Liste aus der textweisen oder satzweisen Markierung stammt, solange der Schwellwert angepasst wird. Die Resultate sind aber mit satzweiser Markierung besser (Kapitel 6).

Die tatsächlich themenbezogenen Wörter können überwiegend im oberen Teil der Rangliste erwartet werden, während weiter unten in der Liste fast nur beliebige Wörter auftauchen. Es könnte also zweckmäßig sein, die Liste an einer gewissen Stelle abzuschneiden und nur den oberen Teil zu verwenden. Dies würde weniger einer verbesserten Satzextraktion dienen, denn die Wörter im unteren Teil der Liste haben nur geringes Gewicht und der Schwellwert wird an die Gewichtsverteilung in der Rangliste angepasst (siehe unten). Es dient aber einer schnelleren Ermittlung des Satzgewichtes, weil sich ein Wortgewicht in der verkürzten Liste schneller auffinden lässt, sowie einer einfacheren Satzkürzung (siehe folgendes Kapitel). In Abschnitt 6.2.3 wird daher die Auswirkung des Abschneidens nach verschiedenen Längen auf die Satzextraktion untersucht. Abschnitt

6.3 behandelt die Untersuchung der Satzkürzung, bei der die Listenlänge ebenfalls eine Rolle spielt.

Die Gewichte der Wörter eines Satzes ergeben in der Summe das Satzgewicht, das über einem gewissen Schwellwert liegen muss, damit der Satz als themenbezogen klassifiziert wird, also in den gezielten Extrakt aufgenommen wird. Die Auswahl des Schwellwertes ist entscheidend für die Qualität der Extrakte. Ein hoher Schwellwert lässt nur wenige Sätze zu, die mit recht hoher Sicherheit auch tatsächlich zum Thema gehören. Dafür werden andere themenbezogene Sätze, deren Gewicht nicht reichte, verpasst. Umgekehrt lässt ein niedriger Schwellwert oft auch Sätze zu, die nicht zum Thema gehören, dafür werden wenige tatsächlich themenbezogene Sätze verpasst. Dieser Sachverhalt wird in den beiden Standardbewertungsmaßen Recall und Precision erfasst.

Recall ist hier der Anteil der themenbezogenen Sätze, die das Verfahren extrahiert. Gibt es beispielsweise 100 themenbezogene Sätze, von denen aber nur 80 über dem Schwellwert liegen, so entspricht dies einem Recallwert von 80 Prozent.

Precision ist hier der Anteil der Sätze über dem Schwellwert, die tatsächlich themenrelevant sind. Liegen 100 Sätze über dem Schwellwert, von denen 80 themenbezogen sind, so entspricht dies einem Precisionwert von 80 Prozent.

Die beiden Maße werden in Abschnitt 6.2.2 allgemein definiert. Ein hoher Schwellwert entspricht also niedrigerem Recall und höherer Precision, während ein niedrigerer Schwellwert besseren Recall, aber schlechtere Precision zur Folge hat. Damit wird deutlich, dass diese beiden Maße im Allgemeinen nicht unabhängig voneinander sind. Welchen Wert man zuvörderst zu maximieren versucht, wird von der beabsichtigten Anwendung abhängen. Beim gezielten Satzfiltern könnte dem Wunsch, auf keinen Fall wichtige Informationen zu verpassen, durch besseren Recall Rechnung getragen werden. Man würde also den Schwellwert heruntersetzen. Andererseits könnte ein Benutzer die Priorität äußern, möglichst selten mit nutzlosen Informationen behelligt zu werden. Zur Erhöhung der Precision kann man dann den Schwellwert heraufsetzen. Es existiert also mit dem Schwellwert ein einfaches Instrument, das Satzfiltern jederzeit an die Bedürfnisse eines Benutzers anzupassen, und zwar noch nach Erstellung der Stichwortlisten. Dem Zwang zum Kompromiss zwischen idealem Recall und idealer Precision kann man jedoch nicht entgehen.

Zu beachten ist, dass der Precision-Wert insofern von der Zusammensetzung der Textsammlung abhängt, als mehr negativ vorklassifizierte Texte mehr Möglichkeiten bieten, Texte oder Sätze irrtümlich positiv zu klassifizieren. Fügt man also zu einer gegebenen Textsammlung Texte hinzu, die nicht das gesuchte Thema behandeln, so bleibt der Recallwert bei Fixierung der anderen Parameter gleich, weil immer noch derselbe Anteil an themenbezogenen Texten bzw. Sätzen erkannt wird. Der Precisionwert könnte aber sinken, weil mehr überflüssige Positivklassifikationen möglich sind. Um diesen Effekt bewerten zu können, kann man den *Fallout*-Wert hinzunehmen ([Lewis 1995]). Dieser ist definiert als Anteil der negativen Beispiele, die das Verfahren irrtümlich positiv klassifiziert. Bei den Auswertungen in Kapitel 6 kommt diese Angabe daher hinzu.

Für den Vergleich zweier Schwellwerte ist ein Mittelwert aus Recall und Precision notwendig. Das sogenannte F_β-Maß (bspw. [Lewis 1995]) erlaubt über einen Parameter

β die Priorisierung von Recall (R) oder Precision (P):

$$F_\beta = \frac{(\beta^2+1)RP}{\beta^2 P + R}$$

F_0 entspricht der Precision und F_∞ dem Recall. Mit $\beta = 0.5$ wird Precision doppelt so stark gewertet wie Recall, $\beta = 1$ wertet beide gleich, und $\beta = 2$ wertet Recall doppelt so stark wie Precision. Für die automatische Ermittlung des besten Schwellwertes in dieser Arbeit werden Recall und Precision gleich priorisiert, weil keine allgemeingültige Benutzerpräferenz vorausgesetzt werden soll (wenngleich wohl zu erwarten ist, dass höherer Recall bei der Versorgung mit wichtigen Informationen den meisten Benutzern wichtiger wäre). Mit $\beta = 1$ wurde also das in der Literatur als Standardmaß verbreitete F_1-Maß verwendet.

Die Ermittlung des besten Schwellwertes nach dem F_1-Wert geschieht automatisch mit Hilfe der Trainingsmenge. Mit Schwellwert 0 erhält man 100% Recall, aber sehr geringe Precision. Der Schwellwert wird schrittweise von 0 an erhöht bis zu einem vorgegebenen Maximalwert; derjenige Schwellwert, der auf der Trainingsmenge zum höchsten F_1-Wert führt, wird ausgewählt. Der Maximalwert für den Schwellwert liegt wegen der Normierung der Satzgewichte (siehe übernächsten Absatz) bei 1, weil kein normiertes Satzgewicht größer als 1 sein kann wegen der Skalierung der Wortgewichte auf den Bereich 0 bis 1. Als Schrittweite ist 0.01 geeignet.

Der sich ergebende Schwellwert ist natürlich abhängig von der Verteilung der Wortgewichte in der Rangliste und damit auch von ihrer Länge. Verschiedene Ranglisten (ob durch verschiedene Verfahren gewonnen oder auf text- gegenüber satzweiser Markierung beruhend) resultieren in verschiedenen Schwellwerten. Ebenso passt sich der Schwellwert an, wenn die Liste nach einem gewissen oberen Teil abgeschnitten wird—es wird dann ein niedrigerer Schwellwert gewählt, weil weniger Wörter in den Sätzen ein Gewicht erhalten.

So wie bisher geschildert, werden bei der Satzauswahl alle Sätze unabhängig von ihrer Länge gleich behandelt; längere Sätze können aber mehr Wörter enthalten, die ein positives Gewicht haben. Um den Schwellwert für alle Sätze vergleichbar zu halten, kann das Satzgewicht durch die Anzahl aller Wörter im Satz geteilt werden (Normierung nach Satzlänge) oder nur duch die Anzahl der Wörter, die überhaupt ein positives Gewicht haben (Normierung nach Gewichtswörtern). Die Normierung nach Satzlänge hat in [Zechner 1996] nicht zu verbesserten Resultaten geführt, während ein ihr ähnliches Vorgehen in [Buyukkokten et al. 2000] die besten Ergebnisse brachte. In dieser Diplomarbeit wurden beide Normierungen ausprobiert, die Normierung nach Satzlänge brachte bessere Resultate (Abschnitt 6.2.3).

Eine weitere Möglichkeit der Unabhängigkeit von der Satzlänge wurde ebenfalls getestet. Das zur Stammformenreduktion verwendete NLP-Werkzeug, Meson, beinhaltet auch einen flachen Parser, der Teilstrukturen in Sätzen erkennt, nämlich Nominal-, Präpositional- und Verbalphrasen (vgl. Kapitel 2). Statt eines Satzgewichtes lassen sich also auch einzelne Phrasengewichte berechnen. Der Satz gilt dann als themenrelevant, wenn mindestens eine seiner Phrasen über dem Schwellwert liegt, welcher genauso ermittelt wird wie zuvor. Die dahinterstehende Idee ist, dass bei längeren Sätzen die interessierende Information nur in einem bestimmten Teil des Satzes stecken könnte, während andere Satzteile, etwa eingeschobene Zusätze, irrelevant sind. Mit dieser Methode sucht man also nach wichtigen Satzteilen statt nach wichtigen Sätzen. Längeren Sätzen entsteht dabei kein Nachteil durch irrelevante Satzteile. Hierbei ist allerdings zu beachten, dass Meson und Wap in den vorliegenden Versionen keine Neben- und Hauptsätze erkennen, sondern nur die erwähnten Teilphrasen. Der Test dieses Vorgehens findet sich in Abschnitt 6.2.3.

Schließlich bietet es sich an, zu testen, ob man ohne die Gewichte der Wörter aus der Stichwortliste auskommt, ob es also genügt, eine ungeordnete Menge von themenbezogenen Wörtern zu benutzen, um Sätze nach dem Thema auszuwählen. Dazu wird die Stichwortliste wie zuvor nach einem gewissen oberen Teil abgeschnitten, in diesem Teil erhalten jedoch dann alle Wörter das Gewicht 1, so dass alle gleich wichtig für die Satzauswahl sind. Den Vergleich der Resultate liefert Abschnitt 6.2.3.

Das in diesem Kapitel erläuterte Verfahren zum gezielten Satzfiltern wird in dieser Diplomarbeit der Klassifikation von Sätzen mit bekannten Textklassifikationsverfahren (Abschnitt 3.3.3) gegenübergestellt. Die Klassifikationsverfahren verwenden dieselbe Markierung derselben Trainingsmenge wie die Ranglistengewinnung. Die in Abschnitt 3.3.3 erwähnten Schwierigkeiten stellen sich der Extraktion über Ranglisten nicht. Diese ist vielmehr flexibler anpassbar an die Bedürfnisse der Benutzer und stellt die gleichen Voraussetzungen an die Markierung der Trainingsmenge.

Als dritte Möglichkeit des Vorgehens wurde schon ein zweistufiges Verfahren erwähnt. Dabei wird zuerst jeder Text als Ganzes klassifiziert, mit einer der beiden Methoden aus Abschnitt 3.3 (SVM oder Zentroidvektor). Die Stichwortliste wird genauso gewonnen wie zuvor, das Satzfiltern wird aber nur auf positiv klassifizierte Texte angewendet. Dies bezeichne ich als *indirektes Satzfiltern*. Da die Klassifikation, nach erfolgtem Training, bei den beiden erwähnten Textklassifikationsmethoden sehr einfach und schnell geht, kann man so Zeit sparen, weil nicht jeder Satz der irrelevanten Texte untersucht werden muss. Dies sollte sich insbesondere dann lohnen, wenn wenige von zahlreich vorliegenden Texten relevant für das interessierende Thema sind.

5. Satzkürzung und SMS-Erstellung

Wie in der Einleitung (Kapitel 1) erläutert wird, ist eine Motivation für diese Arbeit die Erstellung eines Dienstes, der Emails zu SMS-Nachrichten verkürzen kann. Dabei ist es sinnvoll, sich auf bestimmte Inhalte zu konzentrieren, die im mobilen Einsatz relevant sein können, um nicht unterwegs mit irrelevanten Informationen behelligt zu werden. Die bisherigen Kapitel haben deutlich gemacht, wie in Texten nach bestimmten Inhalten gesucht werden kann. Die in dieser Arbeit angewendeten Verfahren werden im vorigen Kapitel beschrieben. Sie liefern zu jedem Text eine—möglicherweise leere—Menge von Sätzen, die als themenbezogen gelten. Mit diesen Sätzen wird die SMS-Nachricht erstellt (ist die gelieferte Menge leer, so wird keine Nachricht erstellt).

Während Emailtexte keiner Längenbeschränkung unterliegen, gilt nach heutigem Standard für SMS-Nachrichten eine strikte Maximallänge von 160 Zeichen (lateinische Buchstaben, Leer- und Sonderzeichen). In der verwendeten Emailsammlung (siehe Abschnitt 6.1.1) beträgt die Durchschnittslänge eines Satzes 78 Zeichen, womit deutlich wird, dass oft die ausgewählten Sätze nicht in eine SMS-Nachricht passen, zumal Angaben zum Absender der Email und zum Betreff sinnvoll sind. Trotz der Reduktion der Nachrichtenlänge durch Satzauswahl kann also eine weitere Verkürzung des Textes angezeigt sein.

Dieses Kapitel stellt die Methoden vor, mit denen die als themenbezogen ausgewählten Sätze weiter verkürzt werden und die fertige SMS-Nachricht zusammengestellt wird. Diese Methoden spielen nur für die erwähnte Anwendung eine Rolle und sind daher losgelöst vom Satzfiltern zu betrachten, das in anderen Bereichen ohne weitere Satzverkürzung angewandt werden kann. Allerdings kann man bei gelungener Verkürzung, also bei einer Verkürzung, die die wesentlichen Satzinhalte in Bezug auf das interessierende Thema unberührt lässt, von verbesserter Informationsextraktion sprechen gegenüber der einfachen gezielten Satzauswahl, weil die interessierenden Informationen kompakter präsentiert werden.

5.1. Satzkürzung

In natürlichen Sprachen ist die Anzahl der Möglichkeiten, einen Sachverhalt auszudrücken, sehr hoch. Vom Sachverhalt selbst werden unterschiedliche Aspekte für unterschiedliche Benutzer mehr oder weniger wichtig sein. Selbst innerhalb eines Satzes sind verschiedene Teile von unterschiedlicher Relevanz. Dies gilt besonders bei frei geschriebenen Texten wie Emails. Bei vielen Sätzen gibt es daher Möglichkeiten, Teile daraus zu entfernen, ohne denjenigen Teil des Inhalts, der von besonderem Interesse ist, zu beeinträchtigen. Dazu ist es nützlich, den grammatischen Aufbau von Sätzen zu betrachten.

Linguistisch gesehen sind Sätze aus Haupt- und Nebensätzen (Halbsätzen) zusammengesetzt. Jeder Halbsatz macht eine eigene inhaltliche Aussage, in deren Zentrum das verwendete Verb steht. Satzteile ohne Verb gelten nicht als Nebensatz, sondern als Adverbiale, also nähere Ausführungen zur Situation des Haupt- oder Nebensatzes, zu dem sie gehören. Jedes Verb bezeichnet einen Zustand oder eine Tätigkeit und hat ein Subjekt, also eine Gruppe von Wörtern, die bezeichnen, wer oder was die Tätigkeit ausführt bzw. für wen der Zustand gilt. Manche Verben erwarten Objekte als Teil ihrer Subkategorisierungen (vgl. Abschnitt 2.2). Verben halten also den Satz in seinem Aufbau zusammen und sind, ebenso wie die Subjekte, wesentliche Elemente, die zum Verständnis des Satzzusammenhangs dienen. Dagegen können Adverbiale oft weggelassen werden, ohne den Satz unverständlich zu machen, wenngleich sich die Aussage dadurch verändern kann.

Diese Beschreibung eines typischen Satzaufbaus ist rein syntaktisch. Aus der Syntax eines gegebenen Satzes, wie sie ein Parser liefert, kann jedoch kein Rückschluss auf seinen Inhalt gezogen werden, auch nicht darauf, an welcher Stelle der wesentliche Inhalt steckt. Zwar sind typischerweise Subjekt und Verb die wichtigsten Elemente, mögen jedoch für das momentane Benutzerinteresse weniger wichtig sein als etwa eine adverbiale Bestimmung der Zeit. Die folgenden Beispiele sollen verdeutlichen, dass eine Information, hier das Ausfallen eines Termins, an syntaktisch verschiedenen Stellen eines Satzes auftreten kann. (Alle Beispiele dieses Kapitels, soweit nicht anders gekennzeichnet, basieren auf Sätzen aus der verwendeten Emailsammlung, siehe Abschnitt 6.1.1.)

(1) `Am Freitag entfaellt die Pruefung um 9.00 Uhr.`

(2) `Am Freitag kann ich auf keinen Fall kommen wegen Zahnarzt und anderer Termine.`

(3) `Leider bin ich am 29. verhindert.`

(4) `Meine Absage für Samstag ist nicht vermeidbar.`[1]

Die Information, dass ein Termin ausfällt, steckt in Beispiel (1) im Verb, in (2) in einem adverbialen Zusatz (`auf keinen Fall`), in (3) im sogenannten Subjektattribut `verhindert` und in (4) im Subjekt des Satzes. Ohne semantische Hinweise besteht also keine Möglichkeit, zu erkennen, welche Elemente eines Satzes weniger wichtig sind als andere, um sie zu entfernen.

Rein syntaktische Satzkürzung wurde für die japanische Sprache vorgeschlagen, unter anderem zur Verkürzung von laufenden Untertiteln bei Fernsehnachrichten, denen hörgeschädigte Zuschauer dann besser folgen können. In [Oguro et al. 2000] wird ein Satz als lineare Folge von Phrasen betrachtet, zwischen denen grammatikalische Abhängigkeiten bestehen. Es wird eine japanische Dependenzgrammatik benutzt, um die Abhängigkeiten zu finden. Für eine Einführung in solche Formalismen siehe beispielsweise [Tarvainen 1981]. Ein Satz erhält in dieser Arbeit ein Gewicht aus der Summe seiner Phrasengewichte plus einem Gewicht, das die Wohlgeformtheit des Satzes im Sinne der verwendeten Grammatik wiederspiegelt. Es beruht auf der Stärke der Abhängigkeiten zwischen den Phrasen, die wiederum aus der Entfernung zwischen abhängigen Phrasen

[1]Konstruiertes Beispiel

und weiteren morphologischen Informationen berechnet wird. Eine Phrase erhält ein Gewicht durch ihre inhaltliche Relevanz. Dies ist nicht rein syntaktisch, aber die Autoren verwenden (in Erwartung besserer zukünftiger Alternativen) die Wortarten in der Phrase, um das Phrasengewicht zu ermitteln, bleiben also auf der Ebene der Syntax. Schließlich wird das Satzgewicht für alle echten Subfolgen der ursprünglichen Phrasenfolge berechnet und für eine vorgegebene Länge (Anzahl von Phrasen) die höchstgewichtete Folge ausgewählt, so dass ein nach diesen Kriterien optimal verkürzter Satz entsteht. Eine Bewertung des Verfahrens wird nicht geliefert.

In einer anderen Arbeit, [Ohtake und Masuyama 2001], werden nur modifizierende Elemente von Nominalphrasen entfernt. Nominalphrasen werden typischerweise von Adjektiven und Relativsätzen modifiziert. Im Japanischen führen die Autoren zehn verschiedene mögliche Muster von Modifizierungen an; die meisten japanischen Nominalphrasen weisen genau zwei davon auf. Alle Modifizierer zu entfernen, wäre zu radikal, weshalb nach selbsterstellten Regeln, die auf der syntaktischen Form der Modifizierer beruhen, nur bestimmte Modifizierer gekürzt werden. Die Heuristik geht dahin, längere und komplexere Modifizierer zu erhalten, da die Nominalphrase sonst nicht mehr in den Satzzusammenhang eingeordnet werden kann. Da nur wenige Satzelemente entfernt werden, verkürzt sich ein durchschnittlicher Text mit dieser Methode um 91%, weshalb für die Erstellung von Zusammenfassungen (Extrakten) eine Satzauswahl vorangestellt wird. Solche Zusammenfassungen werden in der Arbeit extrinsisch bewertet, es fehlt jedoch der Vergleich mit Extrakten ohne die Kürzung von Nominalphrasen.

Für diese Diplomarbeit müssen andere Methoden angewandt werden. Das Verfahren zum Satzfiltern aus dem vorigen Kapitel verwendet gewichtete Wortlisten, die aus der Markierung einer Beispielsammlung gewonnen werden. Wie schon erläutert wurde, gelten diese Wortlisten als Repräsentation des interessierenden Themas, weshalb die Satzauswahl auf ihnen beruht. Das Gewicht der Wörter aus der Liste spiegelt dabei wieder, wie stark ein Wort zum Thema gehört. Es liegt also nahe, wenn Kürzung notwendig ist, nur Satzteile zu entfernen, in denen keine themenbezogenen Wörter auftauchen. Damit bleiben diejenigen Teile eines Satzes, die dafür gesorgt haben, dass er ausgewählt wurde, erhalten und es kann davon ausgegangen werden, dass in ihnen die wichtige Information steckt. Durch die Wortlisten werden also die entscheidenden semantischen Hinweise zur Kürzung gegeben, auch wenn dies weit entfernt von einer echten semantischen Analyse des Satzes ist.

In dieser Arbeit wird dazu so vorgegangen, dass die automatisch erstellte Wortliste nach einem gewissen oberen Teil abgeschnitten wird, mit der Erwartung, dass dieser Teil dann hauptsächlich themenbezogene Wörter aufweist. Wo abgeschnitten werden sollte, wird in Abschnitt 6.3 untersucht. Eine andere Möglichkeit ist die Ermittlung eines weiteren Mindestgewichtes für Satzteile, um der Kürzung zu entgehen. Dies wird bei einem anderen Experiment verwendet, dass in Abschnitt 5.3 beschrieben wird.

Mit Berücksichtigung des obigen linguistischen Wissens können allerdings auch nicht beliebige Elemente aus Sätzen entfernt werden, ohne ihren Aufbau zu zerstören und sie unverständlich zu machen. Ein Vorteil der Art von Informationsextraktion dieser Diplomarbeit ist es ja gerade, den Satzzusammenhang, in dem die Information ausgedrückt wurde, zu erhalten, so dass Leser daraus weitere Informationen entnehmen können. Satzkürzung sollte also behutsam erfolgen. Die im folgenden Kapitel beschriebenen Experimente bestätigen dies. Wegen ihrer zentralen Stellung im Satz werden also mit den folgenden Methoden zur Satzkürzung Verben nie entfernt. Eine Ausnahme bildet die Entfernung eines ganzen Nebensatzes: Im folgenden Beispiel (5) kann der Nebensatz gestrichen werden, wenn nur die Termininformation von Interesse ist, wenn also kein Wort im Nebensatz ein positives Gewicht hat.

(5) `Der Termin heute um 15.00 Uhr in der Klinik hat sich uebrigens erledigt, fuer den Fall, dass jemand von Euch kommen wollte.`

Mit diesen Überlegungen wird klar, dass aufgrund der strikten Platzbeschränkung in SMS-Nachrichten zwei widerstrebende Interessen ausgeglichen werden müssen: Einerseits sollen so viele der ausgewählten Sätze wie möglich untergebracht werden, um viele Informationen weiterzuleiten; andererseits dürfen die Sätze nicht zu stark gekürzt werden, um nicht unverständlich zu werden. Der Rest dieses Kapitels beschreibt daher ein flexibles Verfahren, bei dem beide Interessen berücksichtigt und je nach Benutzerpräferenzen unterschiedlich gewichtet werden können. Die Untersuchung des Verfahrens anhand eines konkreten Themas, den Terminabsprachen, folgt im nächsten Kapitel.

Die wesentliche Idee besteht darin, Kürzungen stufenweise vorzunehmen, wobei jede Stufe etwas radikaler kürzt als die vorige. Auf den ersten Stufen werden „sichere“ Kürzungen verwendet, also solche, die den Satzinhalt nicht verändern. Auf den höheren Stufen werden dann ganze Phrasen gestrichen, wie sie durch den Parser geliefert werden, wenn sie kein Gewichtswort enthalten. Da der erste eingesetzte Parser, in MESON eingebaut, eine zum Teil nicht zufriedenstellende Phrasenerkennung hat, wurde zusätzlich WAP eingesetzt. Die Kürzungen in dieser Diplomarbeit beruhen auf der Ausgabe von WAP, allerdings war dies nur möglich, indem die WAP-Ergebnisse für jeden Satz fest gespeichert wurden. Für den Einsatz „online“ ist WAP erheblich zu langsam: Das Parsen der verwendeten Textsammlung (Abschnitt 6.1.1) dauert mit WAP etwa 54 Stunden, mit MESON dagegen 43 Minuten auf einer SUN U10 Workstation mit 440 MHz Prozessortakt. Diese Arbeit soll jedoch auch untersuchen, welche Qualität mit guten sprachverarbeitenden Werkzeugen erreicht werden kann, weshalb der Einsatz von WAP gerechtfertigt ist. MESON erreicht seinen Geschwindigkeitsvorteil durch die Verwendung endlicher Automaten; es spricht prinzipiell nichts dagegen, durch Verbesserung der Automaten eine ähnlich gute Phrasenerkennung zu erhalten wie bei WAP. Mit WAP werden mehr und längere Wortfolgen als Phrase erkannt; die durchschnittliche Anzahl der erkannten Phrasen pro Satz in der genannten Textsammlung ist 4.7 mit WAP gegenüber 6.9 mit MESON. In vielen Fällen läuft die Verwendung von MESON daher auf eine etwas stärkere Kürzung hinaus, die den Satz weniger verständlich macht.

WAP liefert eine syntaktische Analyse des Satzes durch Teilphrasen, von denen einige entsprechend der Funktionsweise von WAP, die in Abschnitt 2.2 beschrieben ist, nicht komplettiert sind. Hier hat also ein Wort Erwartungen ausgelöst, die nicht erfüllt werden konnten. Solche Phrasen werden zuerst gekürzt vor den komplettierten Phrasen, weil davon auszugehen ist, dass die komplettierten Phrasen eine deutlichere Struktur haben, die die Lesbarkeit des Satzes erhöht. Aus der Ausgabe von WAP wird stets die längste gefundene Phrase verwendet, auch wenn sie untergeordnete Phrasen enthält. Da die zur Verfügung stehende WAP-Software noch nicht völlig ausgereift ist, konnten einige Sätze

Stufe	Art der Kürzung
0	Keine Kürzung
1	Abkürzungen vornehmen
2	Anrede und Grußformel entfernen
3	Füllwörter entfernen
4	Artikel entfernen
5	Adjektive ohne Gewicht entfernen
6	Eingeklammerte Teile entfernen
7	WAP-Adverbphrasen entfernen
8	Offene Präpositionalphrasen ohne Gewicht entfernen
9	Komplettierte Präpositionalphrasen entfernen
10	Offene Nominalphrasen entfernen
11	Komplettierte Nominalphrasen entfernen
12	Wortarten-Positivliste anwenden
Sonderstufe	Nebensätze ohne Gewicht entfernen

Tabelle 5.1.: Übersicht über die Kürzungsstufen. Je höher die Stufe, desto radikaler ist die Kürzung. Genauere Erläuterungen jeder Stufe finden sich im Text.

des Korpus nicht damit geparst werden; bei ihnen wurde auf die Ausgabe von MESON zurückgegriffen.

Die Kürzungsstufen im Einzelnen finden sich in der Übersicht in Tabelle 5.1 und werden im folgenden kurz erläutert.

1. Auf Stufe 1 werden gängige Abkürzungen vorgenommen. So wird `gegebenenfalls` durch `ggf.` ersetzt und ähnliches[2]. Insbesondere werden auch die Wochentage durch ihre zweibuchstabigen Abkürzungen ersetzt, also `Montag` durch `MO` usw.

2. Wenn es sich um den ersten oder letzten Satz einer Email handelt, werden Anrede oder Grußformel entfernt, da sie wohl selten zur wichtigen Information zählen und der Absender der Email ohnehin genannt wird (siehe den folgenden Abschnitt 5.2). Zur Erkennung dienen einfache Muster.

3. Als Füllwörter zählen Wörter, die in den meisten Fällen vergleichsweise inhaltsleer sind, zum Beispiel `mal`, `naja`, `überhaupt`, `halt`, `eben` usw. In manchen Verwendungen tragen diese Wörter natürlich eine wichtige Bedeutung, die dann durch die Kürzung verlorengeht. Allerdings kommt zum Beispiel das Wort `eben` in der verwendeten Emailsammlung nie als Adjektiv in der Bedeutung „flach" vor, sondern nur als Satzadverb wie in `Dann komm ich eben um 10 Uhr`. Eine komplette Liste der Füllwörter findet sich im Anhang, Abschnitt A.1 auf Seite 92.

4. Artikel wie `der`, `das`, `ein` tragen wenig zum Satzverständnis bei und können fast immer problemlos entfernt werden.

[2]Hier werden viele der von MESON vorgenommenen Abkürzungsersetzungen (vgl. Abschnitt 2.1) wieder rückgängig gemacht.

5. Die meisten Adjektive stehen vor Nomen, so dass die Nomen und damit der entsprechende Satzteil zumeist auch noch verständlich sind, wenn das Adjektiv fehlt. Nur Adjektive, die nicht in der Wortliste stehen, werden entfernt.

6. Die Einklammerung von Wörtern oder Satzteilen zeigt meistens gerade deren geringere Relevanz an.

7. Wap erkennt Wörter wie `zusammen`, `sonst` oder `zwar` als einwortige adverbiale Phrasen, die auf dieser Stufe entfernt werden.

8. Ab dieser Stufe beginnt die Kürzung ganzer Phrasen. Nur Phrasen, die kein Wort aus der Themenwortliste enthalten, werden entfernt. Präpositionalphrasen bilden weit häufiger als Nominalphrasen adverbiale Zusätze, während die Nominalphrasen Subjekt und Objekt des Satzes bilden können. Deshalb werden Präpositionalphrasen zuerst gekürzt. Wie oben erläutert, werden nicht komplettierte Phrasen zuerst entfernt.

9. Daran schliessen sich die komplettierten Präpositionalphrasen an.

10. Ganze Nominalphrasen zu entfernen, ist bereits ein recht radikaler Schritt. Nicht immer erhalten alle wichtigen Nominalphrasen auch ein Gewicht aus der Rangliste.

11. Die komplettierten Nominalphrasen folgen wieder auf die nicht komplettierten.

12. Als radikalste Kürzung kommt danach noch in Betracht, alle Wörter zu entfernen, die keine zentrale Wortart wie Nomen, Verb, Präposition oder Negation haben. Die meisten dieser Wörter wurden jedoch schon auf den vorigen Stufen entfernt.

* Die Sonderstufe „Nebensätze entfernen“ wird unabhängig von den anderen Stufen eingesetzt. Dies wird unten beschrieben.

Die verschiedenen Stufen bieten die notwendige Flexibilität bei der SMS-Erstellung, wie der nächste Abschnitt zeigen wird. Zudem kann durch die Angabe einer maximalen Stufe eine zu radikale Kürzung verhindert werden. Bis zur Stufe sechs kann man davon ausgehen, dass die vorgenommenen Kürzungen recht sicher sind, also die Lesbarkeit wie den Inhalt des Satzes nicht verändern. In Einzelfällen stimmt dies jedoch nicht.

Manche Stufen finden nur bei wenigen Sätzen Anwendung, so dass kein Beispielsatz in der Emailsammlung existiert, der alle Stufen verdeutlichen könnte. Es folgen einige Beispiele für die Wirkung der Stufen anhand mehrerer Sätze. Um den Lesern der verkürzten Nachrichten anzuzeigen, dass Kürzungen erfolgten, wird das Zeichen `^` eingesetzt; es steht für ein oder mehrere ausgelassene Wörter.

Der erste Beispielsatz illustriert die ersten fünf Stufen. In (6) ist der ursprüngliche Satz angegeben, darauf folgt das Ergebnis nach jeder der Stufen eins bis fünf.

(6) `Hallo, wie wäre es denn mit einem gemeinsamen Lunch am Montag?`

(7) `Hallo, wie wäre es denn mit einem gemeinsamen Lunch am MO?`

(8) `^wie wäre es denn mit einem gemeinsamen Lunch am MO?`

(9) `^wie wäre^denn mit einem gemeinsamen Lunch am MO?`

(10) `^wie wäre^denn mit^gemeinsamen Lunch am MO?`

(11) `^wie wäre^denn mit^Lunch am MO?`

Das Wort `es` wird zu den Füllwörtern gerechnet (und daher auf Stufe drei gestrichen), weil es meistens als Subjektersatz dient und der Satz auch ohne es verständlich bleibt. Das Wort `denn` ist in diesem Satz wie in vielen anderen ebenfalls ein Füllwort; weil es jedoch auch als Einleitung für eine Begründung vorkommt, steht es nicht auf der Liste der zu streichenden Füllwörter. Das Resultat der ersten fünf Kürzungen (11) ist hier noch gut verständlich.

Ein weiteres Beispiel verdeutlicht die Kürzung von Präpositionalphrasen. In (12) ist der ursprüngliche Satz angegeben und in (13) die Version nach Anwendung der ersten sechs Stufen. Durch Wegfallen der Anrede und einiger nicht so relevanten Adjektive ist der Satz deutlich kürzer, aber immer noch gut verständlich. Beispiel (14) zeigt den Satz nach Entfernung der Präpositionalphrasen.

(12) `Liebe Kollegiaten und Stipendiaten, wegen eines wichtigen unaufschiebbaren Termins im Ministerium in meinem Amt als Prorektorin muss die Kompaktveranstaltung zur Statistik am 05.06.1997 leider ausfallen.`

(13) `^wegen^Termins im Ministerium in meinem Amt als Prorektorin muss^Kompaktveranstaltung zur Statistik am 05.06.1997^ausfallen.`

(14) `^wegen^Termins^muss^Kompaktveranstaltung^am 05.06.1997^ausfallen.`

Auch in (14) ist die wesentliche Information noch erhalten. Die weiteren Kürzungsstufen ändern nichts mehr an (14), weil die Nominalphrase `Termins` ein positives Gewicht hat und `Kompaktveranstaltung` von WAP als Objekt des Verbes `muss` erkannt wurde, also mit zu einer Verbalphrase gehört und Verbalphrasen nicht entfernt werden. MESON erkennt dies nicht und erkennt auch einige Präpositionalphrasen nicht korrekt. Die Version derselben Kürzungsstufe unter Verwendung der Ausgabe von MESON lautet:

(15) `^wegen^Termins^als^muss^zur^am 05.06.1997^ausfallen.`

Während in den bisherigen Beispielen (außer (15)) die Kürzung noch sinnvolle Satzreste lieferte, zeigen die folgenden Sätze, dass manche Kürzungen auch negative Wirkungen haben können. Satz (16) ist die gekürzte Version bis Stufe elf (Entfernung der nichtgewichteten Nominalphrasen) von Satz (17).

(16) `^Konsens ist, dass^Termin^ist, wir beraumen^LS-Tee fuer 10:30 Uhr^, im Besprech-raum, ich denke,^Stunde,^bis zum Mittagessen haben, wird^.`

(17) `Also, allgemeiner Konsens ist, dass der Termin egal ist, wir beraumen also den LS-Tee fuer 10:30 Uhr an, im Besprech-raum, ich denke, eine Stunde, die wir dann bis zum Mittagessen haben, wird reichen.`

Während die Uhrzeit des LS-Tees noch erhalten bleibt, ist der Rest des Satzes in (16) sehr verstümmelt. Das Verb `reichen` wird von WAP als Nomen erkannt und deshalb gekürzt. Die Stufe zwölf ist in den meisten Fällen zu radikal:

(18) `^Konsens^, dass^Termin^,^beraumen^fuer 10:30 Uhr^, im^,^denke, ^Stunde,^bis zum Mittagessen haben, wird^.`

Es ist zu beachten, dass MESON für jedes Wort alle seine möglichen Wortarten liefert, da kein Tagger eingebaut ist. Damit kann es passieren, dass Wörter entfernt werden, die im gegebenen Zusammenhang nicht die Wortart haben, nach denen die aktuelle Kürzungsstufe sucht. Mit einem Tagger könnte zum Beispiel zwischen den verschiedenen Verwendungen von `denn` unterschieden werden.

Das Entfernen ganzer Nebensätze, die kein Gewicht haben, wird deshalb als Sonderstufe behandelt, weil es nicht zwangsläufig als letzte Stufe angewandt werden muss, sondern auch eher schon sinnvoll sein kann. Dies ist einsichtig anhand von Beispiel (5) auf Seite 44: Durch Streichung des Nebensatzes ab `fuer den Fall, ...` wird dieser Satz bereits um die Hälfte seiner Länge reduziert, ohne dass die Information über den Terminausfall verloren geht. Wenn die so gekürzte Version bereits kurz genug ist, wäre es unsinnig, eine der anderen Stufen anzuwenden, durch die die Verständlichkeit nur sinken kann. Würde man umgekehrt erst alle zwölf Stufen durchgehen, könnte der Satz schon unlesbar werden, bevor man den Nebensatz wegnimmt. Andererseits könnte die Entfernung eines ganzen Halbsatzes überflüssig sein, wenn durch wenige Kürzungen der niedrigen Stufen der Satz bereits kurz genug wird.

Wann sollte also die Halbsatzentfernung durchgeführt werden? Es kann anhand der verwendeten Emailsammlung festgestellt werden, dass die Anwendung der Stufen eins bis zwölf (ohne Halbsatzentfernung) die Sätze der Sammlung, die bei der Satzextraktion ausgewählt werden[3], um durchschnittlich die Hälfte verkürzt, gemessen in der Anzahl der Zeichen (inklusive Leerzeichen zwischen den Wörtern; siehe Abschnitt 6.3.1). Wenn also bekannt ist, um welchen Faktor ein Satz gekürzt werden muss, und dieser Faktor unter 1/2 liegt, so ist eine Halbsatzentfernung im Durchschnitt der Fälle sinnvoll, weil die notwendige Kürze mit den anderen Stufen allein nicht erreicht werden kann. Sie wird also in Abhängigkeit von der gewünschten Ziellänge durchgeführt, und zwar nach Stufe zwei, weil die ersten beiden Stufen die Satzaussage sicher nicht berühren. Dieses Vorgehen findet auch bei der SMS-Erstellung Verwendung, wie im nächsten Abschnitt beschrieben wird. Alternativen dazu, etwa die unbedingte oder spätere Anwendung der Sonderstufe, sind leicht realisierbar.

Da MESON und WAP keinen Satzparser beinhalten, der eine Unterteilung in Haupt- und Nebensätze vornimmt, wurde die Halbsatzentfernung mit Heuristiken implementiert. Halbsätze werden oft durch Kommata und Konjunktionen (`und`, `aber`, `dass` usw.) getrennt. Jeder Halbsatz muss ein Verb enthalten. Daher werden zur Halbsatzerkennung die Wortfolgen zwischen Kommata bzw. Konjunktionen betrachtet; diejenigen, die ein Verb enthalten, gelten als Halbsatz. Eine Überprüfung dieses Vorgehens anhand der Emailsammlung erbrachte, dass es nicht zu grob ist, also nie zwei Halbsätze zu einem

[3] Die anderen Sätze werden stärker gekürzt, da sie weniger oder keine Wörter aus der Stichwortliste enthalten.

zusammenfasst, denn kein so gefundener Halbsatz enthält mehr als ein Hauptverb. Wegen des fehlenden Taggers entstehen jedoch Fehler durch falsche Erkennung von Verben, außerdem dienen Konjunktionen auch der Verbindung anderer Satzelemente. Mit späteren Versionen von MESON lässt sich dieses Problem besser behandeln, da ein Satzparser integriert sein wird. Die Ergebnisse des obigen Vorgehens sind jedoch auch zufriedenstellend.

Zuletzt sei herausgestellt, dass die besprochenen Kürzungen von der Wortrangliste abhängig sind, da nur Wörter und Phrasen ohne Gewicht gekürzt werden. Die Bestimmung der besten Verfahren zur Erstellung der Wortranglisten (Kapitel 6) bezieht jedoch die Qualität der sich ergebenden Kürzungen nicht mit ein, da diese nur sehr schwer messbar ist. Diese Qualität wird durch die Lesbarkeit und Verständlichkeit sowie den Informationsgehalt der gekürzten Sätze bestimmt und wird gesondert bewertet, wie ebenfalls in Kapitel 6 erläutert wird (Abschnitt 6.3.2).

5.2. SMS-Erstellung

Dieser Abschnitt behandelt das Vorgehen zur Erstellung einer SMS-Nachricht aus einer Menge von ausgewählten Sätzen einer Email, unter Zuhilfenahme des Kürzungsinstrumentariums aus dem vorigen Abschnitt. Tatsächlich sind die ausgewählten Sätze nicht als Menge gegeben, sondern mit zwei unterschiedlichen Reihenfolgen versehen: Zunächst mit der Reihenfolge, mit der sie in der Email vorkamen, und zweitens mit dem Rang, der sich durch ihr Gewicht ergibt. Ein Satzgewicht berechnet sich aus der Summe der Wortgewichte, auf bestimmte Weise nach der Länge normiert (Abschnitt 4.3). In Sätzen mit höherem Gewicht wird mehr bzw. wichtigere Information vermutet, daher sollten diese Sätze vorrangig behandelt werden.

Außer den gegebenen Sätzen sollte auch Information zum Absender der Email untergebracht werden sowie die Betreffzeile. Beide liefern für den Empfänger wichtige „Meta-Informationen“, stellen also den Zusammenhang her, innerhalb derer die Nachricht interpretiert werden muss und ohne den sie oft nicht verständlich sein kann. Da der Platz in einer SMS sehr begrenzt ist (160 Zeichen), wird für diese Arbeit nur der erste Teil der Emailadresse des Absenders verwendet, nämlich der Teil vor dem `@`. Aus dem gleichen Grund werden von der Betreffzeile nur maximal 20 Zeichen verwendet. Lautet die Absenderadresse beispielsweise `euler@uni-dortmund.de` und der Betreff `Diplomarbeit fast fertig`, so beginnt die SMS-Nachricht mit `euler(Diplomarbeit fast fe):`, wofür 29 der 160 zur Verfügung stehenden Zeichen (inklusive einem Leerzeichen nach dem Doppelpunkt) verbraucht werden.

In vielen Fällen liefert die Satzauswahl nur einen oder zwei Sätze, die kurz genug sind, um hinter die Absender- und Betreffinformation zu passen. Oft müssen die Sätze jedoch erst gekürzt werden. Dies geschieht stufenweise, bis sie passen, so dass sie stets nur so stark gekürzt werden, wie es notwendig ist. Wenn sie nach allen Kürzungen immer noch zu lang sind, können nicht alle Sätze komplett untergebracht werden. In diesem Fall ist es sinnvoll, die höchstgewichteten Sätze unterzubringen, aber in der Reihenfolge, in der sie ursprünglich vorkamen. Der letzte Satz kann dabei unter Umständen nicht vollständig untergebracht werden.

Auf diesen Überlegungen und den Ausführungen zur Halbsatzentfernung im vorigen

Eingabe: Nach Gewicht sortierte Sätze mit Kennzeichnung ihrer ursprünglichen Reihenfolge, sowie ein Informationsstring für den Anfang der SMS
Ausgabe: Ein Text mit Maximallänge 160 Zeichen, der mit dem Informationsstring beginnt

1. Berechne g, die Gesamtlänge der gegebenen Sätze in Zeichen
2. Bestimme den notwendigen Kürzungsfaktor $r = 160/g$
3. Für jede Kürzungsstufe $x = 0$ bis 12:
   a) Kürze alle Sätze bis zur Stufe x; falls $r < 1/2$ und $x > 2$, entferne Halbsätze nach der zweiten Stufe
   b) Bilde eine SMS-Nachricht mit den gekürzten Sätzen, nach dem Algorithmus aus Abbildung 5.2
   c) Bestimme, welcher Anteil an den gekürzten Sätzen in der Nachricht untergebracht werden konnte (wortweise gemessen). Falls der Anteil 1 ist, Schleife abbrechen
4. Rückgabe der SMS-Nachricht

Abbildung 5.1.: Algorithmus zur Erstellung einer SMS-Nachricht, äußerer Teil.

Abschnitt beruht das Vorgehen zur Erstellung einer SMS-Nachricht, das am übersichtlichsten in Form eines Algorithmus angegeben werden kann, siehe Abbildungen 5.1 und 5.2. Der äußere Teil des Algorithmus (Abbildung 5.1) kürzt die Sätze stufenweise so lange, bis alle Sätze in der SMS untergebracht werden konnten oder alle möglichen Kürzungen vorgenommen wurden. Dabei wird die Halbsatzentfernung so verwendet wie im vorigen Abschnitt erläutert. Im inneren Teil (Abbildung 5.2) wird ermittelt, wieviele der ranghöchsten Sätze in den Text passen. Diese werden anschließend in ihrer ursprünglichen Reihenfolge des Vorkommens im Original in den Text gefügt (Schritt 6); dabei wird zusätzlich das Sonderzeichen # zwischen zwei Sätzen eingefügt, wenn Sätze aus dem Ursprungstext dort übersprungen wurden durch die Satzauswahl (nicht in der Abbildung erwähnt). Der letzte Satz in der Gewichtsreihenfolge, der nicht vollständig in den Text aufgenommen werden kann, wird bei diesem Verfahren immer am Ende der Nachricht platziert, damit nicht ein unvollständiger Satz von vollständigen gefolgt wird (Satz u in Abbildung 5.2). Damit kann der SMS-Text Sätze in einer Reihenfolge aufweisen, die nicht dem ursprünglichen Vorkommen in der Email entspricht; dies gilt jedoch stets nur für den letzten, unvollständigen Satz. Hier sind auch andere Verfahren denkbar, zum Beispiel die ursprüngliche Reihenfolge strikt einzuhalten und den nach dieser Reihenfolge letzten Satz unvollständig zu belassen. Da dies jedoch der höchstgewichtete Satz sein könnte, wurde das angegebene Verfahren gewählt. In dieser Version ist die Laufzeit des Algorithmus wegen der Sortierung der Sätze in Schritt 4 des inneren Teils beschränkt durch $O(km \cdot n \log n)$, wenn k die Anzahl der Kürzungsstufen, n die Anzahl der Sätze und m die Länge des längsten Satzes in Worten ist.

Der Algorithmus kann modifiziert werden, indem im äußeren Teil (Abbildung 5.1) in Schritt 3c) nicht gefordert wird, dass der Anteil der untergekommenen Sätze 1 ist. Ein Anteil von 9/10 etwa könnte bereits zufriedenstellend sein, da damit 90% der Wörter der ausgewählten Sätze im SMS-Text untergekommen sind. Außerdem muss die maximale

Eingabe und *Ausgabe:* wie in Abbildung 5.1

1. Setze $S = \emptyset$
2. Beginne den Text T mit dem Informationsstring
3. Für alle Saetze s nach absteigendem Gewicht:

   Wenn s vollständig in den Text T passt, so hänge s an T und füge s zur Menge S; sonst setze $u = s$ und beende die Schleife
4. Sortiere die Sätze in S nach ihrer ursprünglichen Reihenfolge
5. Lösche den Text T und beginne ihn wieder mit dem Informationsstring
6. Hänge die neu sortierten Sätze an den Text T, hinter den Informationsstring
7. Hänge soviel wie möglich von Satz u an T, bis die Maximallänge von 160 Zeichen erreicht ist
8. Rückgabe von T

Abbildung 5.2.: Algorithmus zur Erstellung einer SMS-Nachricht, innerer Teil.

Kürzungsstufe nicht 12 sein (Schritt 3 in Abbildung 5.1), wovon bei der Auswertung der Verfahren in Kapitel 6 auch Gebrauch gemacht wurde.

Mit dem beschriebenen Verfahren lässt sich die Abwägung zwischen Unterbringung von möglichst viel Information und Lesbarkeit der Ergebnistexte (vgl. Seite 44) gezielt vornehmen. Wenn man bei der Lesbarkeit sichergehen möchte, kann man die maximale Kürzungsstufe herabsetzen. Dafür wird man bei längeren Texten nicht alle Sätze in der SMS unterbringen können. Geht es dagegen nur um nackte Information, so kann auch eine radikale Satzkürzung diese meist nicht entfernen, wenn die informationstragenden Wörter in der Stichwortliste stehen. Beispielsweise sind Zeitangaben in der getesteten Termindomäne (Kapitel 6) sehr wichtig und werden dementsprechend so gut wie nie entfernt. Allerdings sind Zeitangaben völlig ohne erklärenden Zusammenhang auch sinnlos. Die Auswertung verschieden stark gekürzter Texte im folgenden Kapitel zeigt, dass eine moderate Kürzung mehr Gewinn für die Leser bringt, auch wenn dann Teile der Information nicht in den Text aufgenommen werden können.

Zuletzt sei hier ein Beispiel für eine Email angeführt, die mit Satzfiltern nach Terminabsprachen sowie Satzkürzung zu einer SMS gekürzt wurde. Die Email lautet:

```
From: kupferstecher@noel.cs.uni-freiland.de
Betreff: Nächstes Treffen und Pacman Demo


Liebe A4lerinnen und A4ler, das nächste Treffen findet am 18.10.2000 im
Raum Campus Süd, GB IV, R. 110 um 10 Uhr statt. Hiermit möchte ich um
Vorschläge für die Tagesordnung bitten. Vorher würde ich gerne die
gwünschte Demonstration von Pacman durchführen, sofern Herr Zeisig und
Beatrix vorher schon Zeit haben 9 Uhr 30 als Starttermin für die Demo
sollte ausreichen. Viele Grüße, Maria
```

Die folgende SMS wurde mit Satzkürzung bis zur Stufe sechs erstellt:

```
kupferstecher(Nächstes Treffen und): ^Treffen findet am 18.10.2000 im
Raum Campus Süd, GB IV, R. 110 um 10 Uhr statt.#^würde ich^gwünschte
Demonstration von Pac
```

Mit Satzkürzung bis zur Stufe zwölf passt eine weitere Zeitangabe in die SMS, deren Verständlichkeit jedoch gesunken ist:

```
kupferstecher(Nächstes Treffen und): ^Treffen findet am 18.10.2000
im^IV,^. 110 um 10 Uhr statt.#^würde^durchführen, sofern^haben 9 Uhr 30
als Starttermin^sollt
```

5.3. Satzkürzung ohne Filtern

Da die Satzkürzung auf den gleichen Wortlisten beruht wie das gezielte Satzfiltern, bietet sich als zusätzliche Möglichkeit an, das Satzfiltern zu überspringen und Texte direkt nur mit dem Satzkürzungsverfahren zu reduzieren. Dies wird hier allgemein als Alternative zum Satzfiltern betrachtet. Dazu müssen die Kürzungen so vorgenommen werden, dass möglichst viele derjenigen Sätze, die sonst weggefiltert worden wären, nun durch die Kürzung ebenfalls ganz entfallen; andererseits sollten möglichst viele Phrasen aus den markierten Sätzen erhalten bleiben. Die Kürzung erfolgt also hierfür nur auf der Ebene der Phrasen, da die Kürzungen der niedrigeren Stufen nicht ausreichen würden, ganze Sätze zu streichen. Statt eines Mindestgewichtes für Sätze wird ein Mindestgewicht für Phrasen ermittelt; alle Phrasen, die über dem Gewicht liegen, bleiben erhalten, die anderen werden entfernt. Man kann dies auch als Phrasenfiltern betrachten. Es wäre auch hierbei möglich, die Kürzung stufenweise vorzunehmen, indem die Art der Phrasen (Nominal-, Präpositional- oder Verbalphrase) in der gleichen Weise wie in Abschnitt 5.1 berücksichtigt wird. Man würde dazu jeweils einen Schwellwert für jeden Phrasentyp benötigen, und je mehr Phrasentypen gekürzt werden, desto mehr irrelevante Sätze fallen ganz weg, während relevante Sätze nicht stärker als nötig gekürzt würden, wenn vor Erreichen der letzten Stufe die Textreduktion schon stark genug ist. Zur Beurteilung, wann das im Allgemeinen—wenn die Länge der Zusammenfassung nicht, wie bei SMS-Nachrichten, genau vorgegeben ist—der Fall ist, müsste man Testpersonen die entstehenden Texte verschiedener Stufen vorlegen.

Für dieses Experiment stand jedoch im Rahmen dieser Arbeit nicht mehr genug Zeit zur Verfügung, um die skizzierte detaillierte Bewertung vornehmen zu können. Die Bewertung der SMS-Texte in Abschnitt 6.3.2 bezieht sich ausschließlich auf die Verfahren aus den Abschnitten 5.1 und 5.2. Um das Experiment des Kürzens ohne vorhergehendes Filtern zu vereinfachen, wird kein Unterschied zwischen den verschiedenen Phrasentypen gemacht—auch Verbalphrasen werden gleich den anderen behandelt, da jeder ganze Satz mindestens ein Verb enthält und kein Satz ganz wegfallen würde, wenn wie zuvor Verbalphrasen niemals gekürzt würden. Das Gewicht einer Phrase wird analog zum Satzgewicht normiert (vgl. Abschnitt 4.3). Da hier wieder ein Schwellwert den Einfluss der niedrig gewichteten, nicht themenbezogenen Wörter abfängt, braucht die Liste nicht abgeschnitten werden.

Als Grundlage für die Ermittlung eines Mindestgewichtes für Phrasen dient die satzweise Markierung. Der Schwellwert wird auf der Trainingsmenge so ausgewählt, dass möglichst viele Phrasen aus markierten Sätzen erhalten bleiben und möglichst viele aus den anderen Sätzen wegfallen. Der Schwellwert beruht also auf den absoluten Häufigkeiten: ist p_m die Anzahl der Phrasen in markierten Sätzen, deren Gewicht über dem gerade getesteten Schwellwert liegt, und p_n die entsprechende Anzahl in nicht markierten Sätzen, so wird von den möglichen Schwellwerten derjenige ausgewählt, der die Differenz $d = p_m - p_n$ maximiert. Wie zuvor wird dabei der Schwellwert von 0 an schrittweise bis 1 erhöht (wegen der analog zu Sätzen vorgenommenen Normierungen kann das Gewicht einer Phrase nicht über 1 liegen).

Zur Bewertung (Abschnitt 6.4) dienen auch hier die *satzweise* gemessenen Recall- und Precisionwerte sowie die durchschnittliche Häufigkeit von erhalten gebliebenen Phrasen in markierten und nicht markierten Sätzen der Testmengen. Hinzu kommen einige—subjektive—Betrachtungen zur Lesbarkeit und Verständlichkeit der entstehenden Kurztexte.

6. Auswertung

Dieses Kapitel beschreibt die Bewertung der Methoden zur Informationsextraktion durch gezieltes Satzfiltern und Satzkürzung aus den vorigen Kapiteln. Ein wesentlicher Teil der Bewertungen behandelt das gezielte Satzfiltern aus Kapitel 4. Dies ist unabhängig von Satzkürzung und SMS-Erstellung, welche beide auf die Erstellung des Email-zu-SMS-Dienstes abzielen.

Gezieltes Satzfiltern hebt auf ein bestimmtes Thema ab, das für Benutzer des Systems von Interesse ist und zu dem daher Informationen aus Texten ermittelt werden sollen. Für den Email-zu-SMS-Dienst ist dieses Szenario sehr passend, weil der Platz in einer SMS so begrenzt ist, dass normalerweise nicht alle Informationen aus einer Email darin untergebracht werden können. Der Dienst besteht also auch darin, nur bestimmte Emails zu SMS-Nachrichten zu verkürzen (Information Filtering, Kapitel 3) und daraus nur die Teile zu verwenden, die zu einem Bereich gehören, über den auch unterwegs schnell Informationen vorliegen müssen.

Um dafür realistische Tests machen zu können, wurde im Rahmen dieser Diplomarbeit eine Sammlung von Emails mit Terminabsprachen gebildet. Änderungen des persönlichen Terminkalenders können sich auf den Tagesablauf einer Person auswirken und sollten ihr daher unverzüglich bekannt gemacht werden. Viele Terminabsprachen werden per Email geführt. Mit Hilfe einer SMS-Nachricht, die die terminbezogene Information aus einer Email enthält, kann der Empfänger sofort erreicht werden. Für diesen Bereich wurde daher nicht nur das gezielte Satzfiltern ausgewertet, sondern auch die Erstellung der SMS-Nachrichten und ihr Informationsgehalt. Diese Domäne ist der automatischen Verarbeitung vergleichsweise gut zugänglich und wurde bereits mehrfach für Projekte im Bereich Sprachverarbeitung herangezogen, etwa für Verbmobil ([Wahlster 1993]) oder Cosma ([Busemann et al. 1997]).

In Abschnitt 4.1 wird behauptet, dass das Verfahren des Satzfilterns über Wortranglisten für beliebige Themen verwendet werden kann, wobei nur eine markierte Sammlung von Trainingstexten benötigt wird, um ein neues Thema behandeln zu können. Die Hinzunahme oder Änderung eines Themas ist also schnell und einfach durchführbar. Um zu beurteilen, inwieweit diese Behauptung richtig ist, wurde als weiterer Test die Extraktion von Wahlergebnissen aus Nachrichtentexten durchgeführt. Für dieses Gebiet spielt die schnelle Weiterleitung der Information keine Rolle, während die Reduzierung der zu lesenden Textmenge auf der Suche nach Wahlergebnissen sehr hilfreich ist. Dementsprechend wurde hierfür nur das Satzfiltern getestet. Die Domäne wurde beispielhaft ausgewählt für eine beliebige gezielte Informationssuche und soll die Leistungsfähigkeit der verwendeten Methoden prüfen.

Dieses Kapitel beginnt mit der Beschreibung der beiden verwendeten Textsammlungen im folgenden Abschnitt, ergänzt durch die Kriterien zur Markierung sowie einigen

Ausführungen zu nichtsprachlichen und sprachlichen Aspekten der Texte. Der darauffolgende Abschnitt 6.2 beleuchtet die Ergebnisse der Satzextraktion. Die Verfahren zur Ranglistenerstellung werden verglichen und die Methoden zur Satzextraktion, die am Ende von Kapitel 4 beschrieben sind, werden einander gegenüber gestellt. Daraufhin (Abschnitt 6.3) folgen Auswertungen der Verfahren zur Satzkürzung und die Beschreibung der Untersuchung des Informationsgehaltes der terminbezogenen SMS-Nachrichten durch Testpersonen. Im letzten Abschnitt 6.4 wird die direkte Satzkürzung ohne vorhergehendes Filtern bewertet.

6.1. Daten

Dieser Abschnitt beschreibt die beiden Textsammlungen, mit denen die Auswertung der Verfahren aus den vorigen Kapiteln durchgeführt wurde.

6.1.1. Termin-Emails

Für die Experimente mit Emails und SMS-Nachrichten wurde eine Sammlung von 560 deutschsprachigen Emails gebildet, von denen die Hälfte (280) Terminabsprachen enthält. Dabei handelt es sich um Ankündigungen von Veranstaltungen oder Treffen, Einladungen, Vorschläge für Zeitpunkte für Treffen, Absagen und Zusagen. Als Termin wird jede Art von Treffen zwischen Personen angesehen, womit zum Beispiel Zeitvorgaben zur Arbeitsplanung nicht als Termin gelten. Zu den terminbezogenen Emails kommen 280 zufällig gewählte andere Emails hinzu, um das Satzfiltern auch auf nichtrelevanten Texten zu prüfen und um die Berechnung der Wortgewichte realistisch zu halten. Zwar ist es unwahrscheinlich, dass das Emailaufkommen einer gegebenen Person zur Hälfte aus Termin-Emails besteht, aber die verwendeten Methoden sind nicht zu stark abhängig von der Zusammensetzung des Korpus (vgl. Kapitel 4). Bei der Bewertung muss diese Zusammensetzung jedoch berücksichtigt werden, was mit dem Fallout-Maß geschieht (Abschnitt 6.2).

Die Emails wurden mir von Freunden, Bekannten und Mitarbeitern des Lehrstuhls für Künstliche Intelligenz der Universität Dortmund zur Verfügung gestellt. Darunter sind private Schreiben ebenso wie offizielle Ankündigungen von Veranstaltungen. Daher variiert der Sprachstil stark: während sich in den privaten Mails zu Teilen ein sehr salopper Sprachgebrauch findet, der unter anderem die Groß- und Kleinschreibung und Zeichensetzung vernachlässigt, sind die Mails aus dem universitären Bereich eher durchdacht formuliert. Tippfehler finden sich jedoch auch hier, in weitaus höherem Maße als bei redaktionell bearbeiteten Texten, die den meisten wissenschaftlichen Untersuchungen über Textzusammenfassung und Informationsextraktion zugrundeliegen. Auch für diese Besonderheit ist das Verfahren dieser Diplomarbeit, den Satzzusammenhang der Informationen möglichst zu erhalten, vorteilhaft, weil dadurch die durch Tippfehler entstehenden Missverständnisse reduziert werden können. Beispielsweise erleichtert der Zusammenhang die Unterscheidung zwischen Datums- und Zeitangabe bei einer Angabe wie `19.10` in einem Text (bei einer Datumsangabe den Punkt hinter der zweiten Zahl zu vergessen, ist ein sehr häufiger Tippfehler).

Um die Auswirkungen der sprachlichen Unterschiede ein wenig untersuchen zu können, wurden die Emails nach ihrer Herkunft in „privat“ und „geschäftlich“ eingeteilt. In Abschnitt 6.2.3 werden die Ergebnisse in beiden Klassen untersucht. Maßgeblich für die Einteilung einer Email war dabei der Absender, so dass etwa Verabredungen zu einer betrieblichen Weihnachtsfeier als geschäftlich deklariert wurden, weil sich zeigt, dass der Sprachstil sich eher nach den Empfängern als nach dem Inhalt richtet. Alle Emails wurden anonymisiert, indem für Namen von Personen, Firmen, Orten und Produkten sowie für (Email-)Adressen Fantasienamen und -adressen eingesetzt wurden.

In den terminbezogenen Emails wurden die Sätze, die die Termininformation enthalten, markiert. Manche Sätze lassen dabei der Entscheidung, sie zu markieren oder nicht, einen gewissen Spielraum. Als Richtlinie galt dabei, dass vor allem die Angaben zur Art des Treffens und zum Zeitpunkt wichtig sind. Ebenso gehört dazu, ob der Termin ausfällt, verschoben wird oder ähnliches, sowie der Ort des Treffens. Zum Beispiel wurden aber Wegbeschreibungen oder andere nähere Ortserläuterungen nicht markiert, ebensowenig wie gesonderte Angaben zum Teilnehmerkreis; natürlich können solche Angaben zu Sätzen gehören, die aus anderen Gründen markiert sind. Für die nicht terminbezogenen Emails wurden die gleichen Quellen verwendet, ihre Zusammensetzung ist daher ähnlich.

Insgesamt enthält die Sammlung in den 560 Emails 3727 Sätze oder 47632 Wörter. Eine Email enthält im Durchschnitt 6 Sätze, 85 Wörter bzw. 532 Zeichen. Von den 3727 Sätzen sind 481 markiert, das sind 12.9% der ganzen Sammlung oder 1.7 markierte Sätze pro terminbezogener Email. Eine zufällige Klassifikation der Sätze in themenbezogen oder nicht würde also—im Erwartungswert—knapp 13% aller Sätze richtig zuordnen. Ein Satz enthält im Durchschnitt 12 Wörter oder 78 Zeichen. Von den 560 Emails sind 247 privat (44%) und 313 geschäftlich. Die Angaben zur Anzahl der Sätze beruhen auf der automatischen Satzerkennung durch Meson, die nicht alle Satzgrenzen richtig erkennt, wie in Abschnitt 2.1 erläutert ist.

Eine Textsammlung von 560 Texten ist recht klein im Vergleich zu den großen öffentlichen Korpora, mit denen viele der wissenschaftlichen Untersuchungen zu Text Mining durchgeführt wurden. Die verwendeten Verfahren zur Erstellung der Stichwortlisten würden vermutlich von einer größeren Textsammlung profitieren, da sie statistischer Natur sind. Jedoch bieten auch die verwendeten Sammlungen genug Grundlage für Aussagen über die Eignung der untersuchten Verfahren. Die Abhängigkeit der Verfahren von der Größe der Datensammlung wird in Abschnitt 6.2.3 näher betrachtet.

Für Emails ist eine spezielle Vorverarbeitung notwendig. Eine über ein Netzwerk verschickte Email ist eine ASCII-Datei mit einem einfachen Aufbau: auf den Header mit Absender, Empfänger, Betreff, Datum und weiteren Angaben folgt eine Leerzeile und dann der Text (Body) der Email. Die Absenderinformation und die Betreffzeile werden in der Vorverarbeitung extrahiert. Außerdem wird ein etwaiges Unterschriftenfeld vom Ende der Email entfernt. Absenderinformation und Betreffzeile werden, wie in Abschnitt 5.2 geschildert, in die zu erstellende SMS-Nachricht übernommen. Weil bei vielen Emails der Sammlung der Header fehlt, wird für deren Absender und Betreff jeweils der String `<unbekannt>` eingesetzt. Dies ergibt bei der SMS-Erstellung einen „Informationsstring“ von 24 Zeichen Länge, während die Durchschnittslänge der echten Informationsstrings 26 Zeichen beträgt. Damit entsteht durch das Fehlen der Absender- und Betreffinformation quasi kein Vorteil bei der SMS-Erstellung durch vermehrten Platz.

6.1.2. Nachrichtentexte mit Wahlergebnissen

Um das gezielte Satzfiltern einem weiteren Test zu unterziehen, wurde eine zweite—etwas kleinere—Textsammlung gebildet, die aus 200 Nachrichtentexten deutscher Tageszeitungen besteht. Quelle dafür waren die im Internet frei zugänglichen Teile der Archive der Zeitungen Die Welt, Frankfurter Rundschau, Süddeutsche Zeitung und Taz sowie der Email-Nachrichtenservice der Deutschen Welle. Kennzeichnend für diese Sammlung ist, dass alle Texte redaktionell bearbeitet wurden und der Sprachstil vergleichsweise einheitlich ist.

Knapp die Hälfte (93) der Texte enthält Nachrichten über Wahlergebnisse von Parlaments- oder Regierungswahlen in aller Welt. Dabei wurden keine ausführlichen Hintergrundberichte und lange Wahlanalysen verwendet, sondern nur vergleichsweise kurze Übersichtsartikel, wie sie etwa auf Titelseiten vorkommen. Die Sätze, die die Ergebnisse nennen, wurden markiert. Dabei wurden folgende Kriterien angewandt: Außer den Prozentangaben der Stimmenverteilung und der Angabe der Sitzverteilung in Zahlen wurden auch sprachliche Angaben wie `Wahlsieger ist ...`, `... wurde bestätigt` oder `... erlitt Niederlage` markiert. Ebenso kommen Gewinn- und Verlustangaben und Angaben zur Wahlbeteiligung hinzu. Wahlanalysen, Angaben zu Wählerwanderungen, Ergebnisse von Abstimmungen innerhalb eines Parlamentes sowie Kommentare von Politikern wurden dagegen nicht markiert. Wiederum können solche Angaben jedoch in Sätzen auftreten, die aus anderen Gründen markiert wurden. Die Nachrichten ohne Wahlergebnisse wurden aus allen Ressorts der verwendeten Zeitungen zusammengestellt, um ein breites Spektrum als Hintergrundkorpus zu haben.

Die Texte dieser Sammlung sind deutlich länger als die Emails, so dass in den 200 Texten 6103 Sätze und 97634 Wörter enthalten sind, etwa doppelt so viele wie bei den Emails. Davon wurden 622 Sätze markiert, also 10.2% der Sammlung, oder 6.7 pro Text, in dem markierte Sätze vorkommen. Der durchschnittliche Text enthält 30 Sätze oder 488 Wörter.

Ähnlich wie in der Termindomäne die Zeitangaben spielen in diesem Bereich Prozentangaben eine große Rolle. Es finden sich jedoch auch Texte zu Wirtschaftsentwicklungen und der Börse in der Textsammlung, die ebenfalls Prozentangaben enthalten und nicht markiert wurden. Im Abschnitt 6.2.4 wird untersucht, wie stark das gezielte Satzfiltern von Prozentangaben abhängig ist.

6.2. Satzfiltern

In diesem Abschnitt werden die Experimente zur Bewertung des gezielten Satzfilterns und ihre Resultate vorgestellt. Der Hauptteil der Experimente wurde mit den Termin-Emails durchgeführt (Abschnitt 6.2.3). Aus Zeitgründen konnten die Experimente mit den Nachrichtentexten nicht so ausführlich ausfallen (Abschnitt 6.2.4), so dass einige der Voreinstellungen, die bei den Emails erfolgreich waren, ohne zusätzliche Tests übernommen werden mussten. Die Ergebnisse sind dennoch aufschlußreich. Zunächst wird jedoch im folgenden Unterabschnitt zusammengefasst, welche Untersuchungen vorgenommen wurden, woran sich in 6.2.2 die Vorstellung der Bewertungsmaße anschließt.

6.2.1. Zusammenfassung der Untersuchungen

Drei hauptsächliche Methoden zum Satzfiltern werden in den bisherigen Kapiteln entwickelt, wie bereits am Ende von Kapitel 4 angedeutet. Zusammengefasst lauten diese:

a) Klassifikation von Sätzen mit bekannten Textklassifikationsverfahren (Abschnitt 3.3.3)

b) Gezieltes Satzfiltern über Ranglisten von Stichwörtern (Kapitel 4)

c) Indirektes gezieltes Satzfiltern, also Klassifikation von Texten mit den Textklassifikationsverfahren und anschließendes Satzfiltern über die Stichwortlisten in den positiv klassifizierten Texten (Abschnitt 4.3)

Dazu stehen bei den Punkten b) und c) jeweils die verschiedenen Verfahren zur Erstellung der Ranglisten zur Verfügung. Diese sind (Abschnitte 4.2.1 bis 4.2.5):

1. Worthäufigkeit
2. Tf-Idf-Gewichte
3. G^2-Methode
4. Information Gain
5. SVM-Gewichte
6. SVM-Gewichte der nach Tf-Idf wichtigsten Wörter

Bei der Vorstellung des gezielten Satzfilterns in Kapitel 4 werden an einigen Stellen mehrere Möglichkeiten des Vorgehens vorgestellt, so dass die entsprechenden Algorithmen parametrisiert sind. Die Ermittlung der besten Parameterwerte muss durch Experimente erfolgen und ist ein Untersuchungsgegenstand dieser Diplomarbeit. Diese (booleschen) Parameter sind:

- Erstellung der Stichwortliste aus der satzweisen oder textweisen Markierung (Abschnitt 4.1)
- Verwendung von Stammformenreduktion und Stoppwortentfernung oder nicht (Abschnitt 4.2)
- Normierung des Satzgewichtes nach Satzlänge oder nach Anzahl der Gewichtswörter (Abschnitt 4.3)
- Auswahl eines Satzes über sein Satzgewicht oder einzelne Phrasengewichte (Abschnitt 4.3)
- Verwendung der Wortgewichte oder Gleichsetzung aller Wortgewichte (Abschnitt 4.3)

Der Einfluss dieser Parameter wird in Abschnitt 6.2.3 gemessen. Ein weiterer Parameter, der untersucht werden kann, ist die Länge des oberen Teils der Stichwortliste, nach dem sie abgeschnitten wird, um eine präzisere Satzkürzung zu ermöglichen. Dieses Abschneiden für die Satzkürzung ist zwar unabhängig von der Verwendung der Liste für die Satzauswahl, mag jedoch sinnvoll sein. Allerdings sorgt die automatische Ermittlung des Schwellwertes (Abschnitt 4.3) für eine Anpassung an die gegebene Verteilung der Wortgewichte und damit auch an die Länge der Liste. Dies wird ebenfalls in 6.2.3 untersucht.

Sowohl Satzklassifikation als auch Satzfiltern über Stichwortlisten benötigen eine Trainingsmenge, um darauf zu lernen oder die Wortlisten daraus zu gewinnen. Die Größe der verwendeten Trainingsmenge ist also eine weitere Variable, deren Einfluss untersucht werden sollte. Die Bewertung der obigen Methoden muss dagegen auf einer Testmenge erfolgen, die zur Trainingsmenge disjunkt ist. Für diese Diplomarbeit wurde daher so vorgegangen, dass pro Experiment eine Testmenge zufällig aus der verwendeten Datensammlung zusammengestellt wurde und die übrigen Daten zum Training verwendet wurden. Vorgegeben wurde dazu die Wahrscheinlichkeit, mit der ein Text zur Testmenge gehört. Hierfür wurden die Werte 0.1, 0.2 usw. verwendet, was einer erwarteten Testmengengröße von 10, 20 usw. Prozent der Daten entspricht, also beispielsweise bei den Terminen 56 Emails beim Wert 0.1, womit dann die Trainingsmenge 504 Emails enthielte. Je kleiner die Trainingsmenge sein kann, ohne zu wesentlich schlechteren Ergebnissen zu führen, desto besser ist das untersuchte Verfahren für den schnellen Einsatz in beliebigen Bereichen geeignet, denn die Markierung einer Trainingsmenge bedeutet einen erheblichen Aufwand. Außerdem kann so beurteilt werden, wie stark die statistischen Verfahren zur Erstellung der Wortrangliste von der Größe der Textsammlung abhängen, also auch, wie aussagekräftig die Ergebnisse dieser Diplomarbeit in Anbetracht der eher geringen Größe der zugrundeliegenden Textsammlungen sind.

Um zu vermeiden, dass zufällig eine besonders (un-)geeignete Testmenge zusammengestellt wird und für ungerechtfertigt gute (schlechte) Ergebnisse sorgt, wird jedes Experiment zehnmal mit verschiedenen, immer zufälligen Testmengen wiederholt. Dadurch wird auch die geringe Größe einer Testmenge, die 10% der nicht sehr großen Textsammlung enthält, kompensiert. Dieses Standardvorgehen nennt man *Kreuzvalidierung*. Dabei ist es sinnvoll, nicht nur das Durchschnittsresultat der zehn Kreuzvalidierungsrunden anzugeben, sondern auch die (empirische) *Standardabweichung* vom Durchschnitt, aus der ersichtlich ist, wie weit die Resultate um den Durchschnitt gestreut sind. Je weniger Streuung die Resultate aufweisen, desto stabiler und unabhängiger von der Zusammensetzung von Trainings- und Testmenge ist das untersuchte Verfahren. Eine niedrige Standardabweichung ist also wünschenswert. Bei Durchführung von r Kreuzvalidierungsrunden mit dem jeweiligen Ergebnis x_1 bis x_r berechnet sich die Standardabweichung s wie folgt. Sei m der Mittelwert von x_1 bis x_r, dann ist s die Wurzel aus der empirischen Varianz, also der durchschnittlichen quadrierten Abweichung von m:

$$s = \sqrt{\frac{1}{r}\left(\sum_{i=1}^{r}(x_i - m)^2\right)}$$

Im folgenden Abschnitt werden die Maße zur Bewertung der Satzextraktion zusammengefasst. Sie lassen sich sowohl für die Trainings- als auch die Testmenge angeben. Für

	Satz markiert	Satz nicht markiert
Klassifikation positiv	a	b
Klassifikation negativ	c	d

Tabelle 6.1.: Contingency Table der möglichen Übereinstimmung oder Abweichung zwischen Klassifikationsergebnis und Vorgabe.

die Bewertung der Leistungsfähigkeit der Verfahren sind natürlich nur die Testmengenwerte von Bedeutung. Diese dienen zur Beurteilung der im vorigen Abschnitt aufgeführten Möglichkeiten, das Verfahren zu parametrisieren. Jedoch ist ein weiterer Parameter, der Schwellwert zur Satzauswahl, direkt von der verwendeten Stichwortliste und der Verteilung der Wortgewichte in ihr abhängig; diese wiederum ist vom Erstellungsverfahren und von der Zusammensetzung der Trainingsmenge abhängig. Insbesondere kann die Wortrangliste nur Wörter aus der Trainingsmenge enthalten und muss dennoch für die Testmenge verwendet werden, um den Einsatz auf neuen Texten beurteilen zu können. Der Schwellwert wird also, wie in Abschnitt 4.3 beschrieben, bei jedem Trainingslauf neu bestimmt, indem derjenige Schwellwert mit dem besten resultierenden F_1-Wert auf der Trainingsmenge ausgewählt wird. Daher sind auch die Trainingsmengenwerte interessant und werden im Abschnitt 6.2.3 kurz betrachtet.

6.2.2. Bewertungsmaße

In Abschnitt 4.3 wurden bereits einige Bewertungsmaße vorgestellt, mit denen die Satzextraktion beurteilt werden kann. Alle beruhen auf der Sichtweise der Satzextraktion als binärer Klassifikation jedes Satzes, mit den beiden Klassen „relevant“ und „nicht relevant“. Die Klassifikation eines Satzes oder Textes durch ein gegebenes Verfahren kann mit der vorgegebenen Markierung übereinstimmen oder nicht; die vier möglichen Fälle sind in Tabelle 6.1, einer sogenannten *Contingency Table*, mit kleinen Buchstaben benannt. Für eine gegebene Testmenge erhalten die Variablen a bis d konkrete Zahlenwerte, deren Summe die Anzahl der Elemente der Testmenge ist. Daraus ergeben sich die verwendeten Bewertungsmaße (vgl. Abschnitt 4.3), die in Prozent angegeben werden. Deren Erläuterung bezieht sich hier zur Veranschaulichung auf die Satzklassifikation, die Definition anhand der Contingency Table ist jedoch allgemein verwendbar.

- *Accuracy* (A) misst den Anteil der korrekten Klassifikationen:
$$A = \frac{a+d}{a+b+c+d}$$
- *Recall* (R) misst, wieviele der markierten Sätze positiv klassifiziert wurden:
$$R = \frac{a}{a+c}$$
- *Precision* (P) misst, wieviele der positiv klassifizierten Sätze auch markiert sind:
$$P = \frac{a}{a+b}$$

- *Fallout* gibt an, wieviele der nicht markierten Sätze irrtümlich positiv klassifiziert wurden:
$$Fallout = \frac{b}{b+d}$$

Zur anschaulichen Erläuterung von Recall und Precision und deren gegenseitiger Abhängigkeit sowie Fallout siehe Abschnitt 4.3. Hinzu kommt das im selben Abschnitt definierte F_1-Maß, das einen Mittelwert zwischen Recall und Precision angibt.

Zur Accuracy sei angemerkt, dass bei den verwendeten Sammlungen bereits Werte von 87 bzw. gut 89 Prozent erzielt werden können, indem *alle* Sätze negativ klassifiziert werden, weil nur 13 bzw. 10 Prozent der Sätze positiv markiert sind. Die erzielten Accuracy-Werte sollten also deutlich über 90 Prozent liegen, um von guten Ergebnissen sprechen zu können. Recall und Precision haben (ebenso wie die Accuracy) einen Optimalwert von 100 Prozent; angesichts der in bisherigen Arbeiten zur Satzextraktion erzielten Ergebnisse (vgl. Abschnitt 3.5.1) sollten Werte über 70 Prozent jedoch bereits als Erfolg angesehen werden, wenngleich die gezielte Extraktion leichter sein mag als die allgemeine, weil nicht mehr jedes wesentliche Thema des Ausgangstextes repräsentiert werden muss und die Kriterien zur Extraktion schärfer bestimmbar sein mögen. Die Werte der allgemeinen Extraktion (um 70 Prozent in [Mani und Bloedorn 1998], siehe Abschnitt 3.5.1) sollten also mindestens erreicht werden. Für den Fallout ist der Optimalwert 0. Es fehlen dazu Vergleichsangaben bei der vorgestellten Literatur. Da der Wert ein Maß dafür ist, wieviele der irrelevanten Sätze ein Benutzer bei Benutzung des Systems lesen müsste, sollten die Werte unter 10, möglichst sogar unter 5 Prozent liegen; damit wäre eine sehr deutliche Erleichterung bei der Suche nach Informationen gegeben.

Wegen der gegenseitigen Abhängigkeit von Recall- und Precisionwerten ist es nützlich, deren gemeinsame Entwicklung bei Veränderung einzelner Parameter zu betrachten. Für gezieltes Satzfiltern ist der Schwellwert der entscheidende Parameter, über den Recall und Precision gesteuert werden, wie in Abschnitt 4.3 beschrieben ist. Ein niedriger Schwellwert führt zu hohem Recall und niedriger Precision, und umgekehrt. Dies lässt sich anhand eines Graphen veranschaulichen, der die Entwicklung von Recall und Precision gegen den Schwellwert aufträgt. Von Interesse ist dabei der Punkt, an dem Recall und Precision den gleichen Wert annehmen, der sogenannte *Breakeven*-Punkt, um eine Beurteilung der Resultate in einer einzelnen Zahl zu erhalten; zu diesem Zweck dient in den folgenden Abschnitten jedoch hauptsächlich das F_1-Maß.

6.2.3. Termine

Satzklassifikation

Zunächst wird die Satzklassifikation mit den Textklassifikationsmethoden untersucht. Wie in Kapitel 3 erwähnt, wurden in dieser Diplomarbeit zwei bekannte Klassifikationsverfahren eingesetzt: Support Vector Machine (SVM) und Zentroidklassifikation. Für beide Methoden werden gute Ergebnisse zur Textklassifikation in der Literatur berichtet, sie werden daher hier stellvertretend für den Bereich Textklassifikation verwendet. Für die SVM-Versuche setzt diese Arbeit die mySVM-Implementation von Stefan Rüping ein ([Rüping 2000]), mit linearer Kernelfunktion.

	Recall	Precision	F_1	Accuracy
SVM	86.2 ± 7.6	71.1 ± 7.7	77.8 ± 6.9	74.8 ± 6.9
Zentroide	79.6 ± 7.4	94.4 ± 3.5	86.2 ± 5.1	87.5 ± 4.3

Tabelle 6.2.: Ergebnisse der Klassifikation ganzer Emails mit den Textklassifikationsverfahren (in Prozent), unter Verwendung von Stammformenreduktion. Jeder Wert ist der Durchschnitt von zehn Kreuzvalidierungsrunden, mit „±“ ist die Standardabweichung in Prozent über die zehn Runden angegeben.

	Recall	Precision	F_1	Accuracy
SVM	90.7 ± 7.2	65.5 ± 7.6	75.5 ± 5.3	70.3 ± 5.3
Zentroide	34.1 ± 11.9	96.3 ± 4.8	49.2 ± 12.2	67.1 ± 7.2

Tabelle 6.3.: Ergebnisse der Klassifikation ganzer Emails mit den Textklassifikationsverfahren (in Prozent), ohne Verwendung von Stammformenreduktion.

Als erstes sollte überprüft werden, ob die guten Resultate bei der Klassifikation ganzer Texte, die in der Literatur berichtet werden, mit den verwendeten Daten und Implementationen wiederholt werden können. Die Tabellen 6.2 und 6.3 zeigen, dass dies der Fall ist. Darin finden sich die Ergebnisse der Klassifikation ganzer Emails in Prozent auf zufällig nach dem oben beschriebenen Verfahren zusammengestellten Testmengen, wobei ein Text mit der Wahrscheinlichkeit 0.1 in die Testmenge aufgenommen wurde. Die Ergebnisse sind zehnfach kreuzvalidiert, die Standardabweichung über die zehn Runden ist jeweils hinter „±“ angegeben. Eine zufällige Klassifikation der Emails würde im Erwartungswert für jeden Eintrag den Wert 50% erzeugen, da genau die Hälfte der Emails in der Sammlung themenbezogen ist. Die erste Tabelle (6.2) zeigt dagegen die Ergebnisse der lernenden Verfahren bei Einsatz von Stammformenreduktion, die zweite (6.3) die ohne Stammformenreduktion. Zumindest die Zentroidklassifikation mit Stammformenreduktion liegt im Bereich der Resultate aus der Literatur ([Han und Karypis 2000]), während die SVM-Performanz darunter bleibt (vgl. [Joachims 1998]), wohl wegen der geringen Größe der Textsammlung.

Es fällt auf, dass die zentroidbasierte Klassifikation wesentlich verbessert wird durch Einsatz der Stammformenreduktion, während die SVM auch mit den unveränderten Texten gut zurechtkommt. Allerdings sind bei der SVM ohne Stammformenreduktion sowohl Precision als auch Klassifikationsgenauigkeit (Accuracy) deutlich niedriger als mit Stammformenreduktion. Die Verwendung von Stammformen scheint sich also auszuzahlen. Die Standardabweichung ist auch bei den guten Werten recht hoch, die Ergebnisse schwanken also je nach Testmenge vergleichsweise stark.

Alle folgenden Tabellen mit Testmengenwerten enthalten ebenfalls jeweils zehnfach kreuzvalidierte Ergebnisse mit Angabe der Standardabweichung. Ebenso setzt sich eine Testmenge jeweils aus etwa einem Zehntel der ganzen Textsammlung zusammen (durch Verwendung der Wahrscheinlichkeit 0.1, zur Testmenge zu gehören), wenn die Tabellen nicht anders gekennzeichnet sind.

		Recall	Precision	F_1	Accuracy
mit Stammformen	SVM	10.7 $\pm$ 4.9	33.6 $\pm$ 8.6	16.0 $\pm$ 6.4	85.2 $\pm$ 3.1
	Zentroide	84.1 $\pm$ 8.9	58.4 $\pm$ 5.4	68.8 $\pm$ 5.1	90.5 $\pm$ 2.4
mit Originalformen	SVM	12.8 $\pm$ 4.8	34.7 $\pm$ 12.7	18.6 $\pm$ 6.8	86.0 $\pm$ 1.9
	Zentroide	48.8 $\pm$ 5.7	54.6 $\pm$ 8.6	51.0 $\pm$ 4.8	87.6 $\pm$ 2.0

Tabelle 6.4.: Ergebnisse der Klassifikation von Sätzen mit Textklassifikationsverfahren.

Die Testmengen-Resultate der Textklassifikationsverfahren für die Satzklassifikation gibt Tabelle 6.4 an, in der die Ergebnisse mit und ohne Stammformenreduktion zusammengefasst sind. Hier wird deutlich, dass die Satzextraktion mit herkömmlichen Textklassifikationsverfahren zu schwache Resultate liefert, obwohl die Klassifikation mit Zentroiden auf den Stammformen bereits an interessante Werte heranreicht. Mögliche Gründe für das schwache Abschneiden sind in Abschnitt 3.3.3 angesprochen. Für die zentroidbasierte Klassifikation gilt immerhin, dass die Ergebnisse deutlich über den Werten liegen, die mit einer zufälligen Klassifikation zu erreichen gewesen wären (diese liegen, wie oben erläutert, bei 13%, dem Anteil der markierten Sätze). Der SVM gelingt dies nur bei der Precision. Man beachte auch die obigen Erläuterungen zur Accuracy bei der Satzextraktion, wonach 87% Accuracy hier bereits mit trivialen Methoden zu erreichen sind.

Das einfache Verfahren der Textklassifikation mit Zentroidvektoren erweist sich also—bei Verwendung von Stammformenreduktion—auf diesen Daten als robuste und effektive Textklassifikationsmethode, die auch auf sehr kurzen Texten nicht einbricht. Dieses Ergebnis untermauert diejenigen aus [Han und Karypis 2000], wo die Zentroidklassifikation ausführlich getestet wird. SVMs scheinen dagegen zur direkten Satzklassifikation nicht geeignet zu sein.

Gezieltes Satzfiltern

Im folgenden wird gezieltes Satzfiltern mit Hilfe der unterschiedlich erzeugten Stichwortlisten untersucht. Zunächst wird eine Stichwortliste, die mit dem Worthäufigkeits-Verfahren errechnet wurde, vorgestellt (Abbildungen 6.1 und 6.2). Teile einer anderen Liste, die auf der gleichen Trainingsmenge mit dem G^2-Verfahren erstellt wurde, finden sich in Anhang A.2.

Die Stichwortliste aus den Abbildungen hier (zum Thema „Terminabsprachen“) enthält nur Stammformen und basiert auf der satzweisen Markierung der Trainingsmenge. Insgesamt enthält sie 833 Wörter, das ist die Anzahl der verschiedenen Wörter, die im markierten Teil der Trainingsmenge vorkommt. Die ersten 25 Wörter zeigt Abbildung 6.1. Die auf den Bereich 0 bis 1 skalierten Gewichte stehen rechts neben den Wörtern, links steht der Rang.

Die Wörter `time` und `date` sind Platzhalter für Zeit- und Datumsangaben, wie sie von Meson automatisch erkannt werden, zum Beispiel `19.45 Uhr` oder `3.11.2000`. Das vierte Wort der Liste stammt vom Ausdruck `Gebäude IV`, der in den geschäftlichen Emails häufig ist, weil er ein Universitätsgebäude der Universität Dortmund bezeichnet. Das Wort

1	`stattfind`	1.0000	14	`reservier`	0.3636
2	`time`	0.9000	15	`dayofweek`	0.3244
3	`termin`	0.7954	16	`hiermit`	0.2727
4	`iv`	0.7273	17	`huette`	0.2727
5	`uhrzeit`	0.6364	18	`offen`	0.2727
6	`nachmittag`	0.5909	19	`bevorzug`	0.2727
7	`uhr`	0.5097	20	`weihnachtsfeier`	0.2727
8	`date`	0.4959	21	`festleg`	0.2727
9	`nord`	0.4545	22	`raum`	0.2307
10	`gebaeude`	0.4545	23	`bibliothek`	0.2272
11	`ok`	0.4242	24	`treff`	0.2078
12	`statt`	0.3863	25	`treffen`	0.2009
13	`bestaetig`	0.3636			

Abbildung 6.1.: Die ersten 25 Wörter einer Stichwortliste zum Thema „Terminabsprachen“, erstellt mit dem Worthäufigkeitsverfahren.

`dayofweek` ist ein Platzhalter für die Namen der Wochentage, also `Montag`, `Dienstag` usw. Wort Nummer 24 ist die Stammform des Verbes `treffen`, während Nummer 25 die des Nomens `Treffen` ist. Es ist erkennbar, dass der obere Teil der Rangliste tatsächlich viele Terminwörter enthält.

Auch weiter unten finden sich noch Terminwörter, wie Abbildung 6.2 zeigt. Das Wort `monthofyear` ist ein Platzhalter für die Monatsnamen `Januar`, `Februar` usw. Dieses und andere Wörter wie `Party`, `Teilnahme` oder `nah`, das als Stammform von `nächst-` in Zeitausdrücken wie `am nächsten Montag` häufig ist, können mit Treffen von Personen in Bezug gebracht werden. Interessant ist das Wort `ab`, das in Zeitangaben wie `ab 10 Uhr` vorkommt, aber auch in Ausdrücken wie `... sage hiermit ab`. Andere Verfahren zur Erstellung der Wortlisten gewichten zum Beispiel die Präpositionen `am` und `um` wegen ihrer Verwendung in Zeitangaben so, dass sie weit oben in der Liste stehen. Ein Beispiel dafür ist die Liste im Anhang A.2. Die Gewichtung dieser Wörter mag ein Beispiel dafür sein, welche Vorteile sich aus der automatischen Erstellung der Listen ergeben, da man bei einer manuellen Erstellung vielleicht nicht an solche unscheinbaren Wörter gedacht hätte.—Am Ende der Stichwortlisten finden sich keine Terminwörter mehr (siehe A.2).

Im folgenden wird untersucht, welche Resultate sich mit solchen Stichwortlisten erzielen lassen, indem für jeden Satz sein Gesamtgewicht bestimmt und mit einem Schwellwert verglichen wird. In Abschnitt 4.3 ist beschrieben, in welcher Weise sich über den Schwellwert Recall und Precision steuern lassen und wie über diese Werte auf der Trainingsmenge der beste Schwellwert bestimmt wird. Zur Überprüfung dieses Prozesses wird daher zuerst die Entwicklung von Recall und Precision auf der Trainingsmenge untersucht. Abbildung 6.3 zeigt, wie sich Recall und Precision bei wachsendem Schwellwert verhalten, hier beispielhaft für die Worthäufigkeitsliste. Das Bild entspricht genau den Erläuterungen im Abschnitt 4.3: bei steigendem Schwellwert sinkt der Recallwert, während der Precisionwert steigt. Auf dieser Trainingsmenge wurde der höchste F_1-Wert für diese Liste beim Schwellwert von 0.05 erzielt, weswegen dieser Schwellwert für die Test-

150	bekanntgeb	0.0909	163	nah	0.0668
151	gen	0.0909	164	morgen	0.0653
152	lokal	0.0909	165	absprech	0.0606
153	monthofyear	0.0866	166	manuskript	0.0606
154	zusammen	0.0844	167	algorithmus	0.0606
155	frueh	0.0727	168	ab	0.0562
156	party	0.0707	169	mindestens	0.0545
157	hin	0.0682	170	teilnehmer	0.0545
158	ausseh	0.0682	171	ring	0.0545
159	konferenz	0.0682	172	stunde	0.0545
160	lad	0.0682	173	rueckmeldung	0.0545
161	weiterhin	0.0682	174	woche	0.0540
162	teilnahme	0.0682	175	wegen	0.0519

Abbildung 6.2.: Wörter aus dem mittleren Teil der Stichwortliste aus Abbildung 6.1.

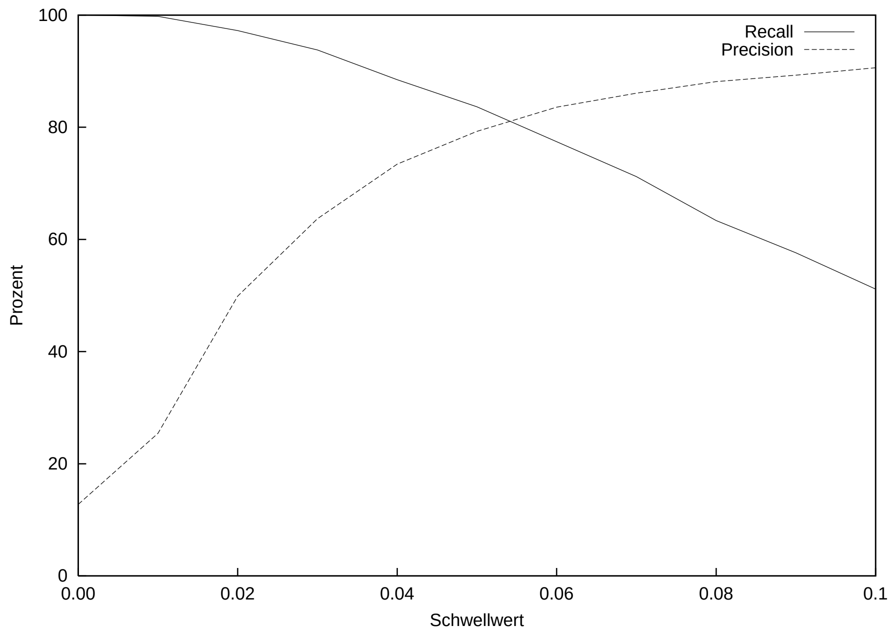

Abbildung 6.3.: Darstellung der Recall- und Precisionwerte bei wachsendem Schwellwert auf der Trainingsmenge, bei der Worthäufigkeitsliste. Der höchste F_1-Wert liegt beim Schwellwert 0.05, der Breakeven-Punkt kurz danach.

Listenerstellung	Recall	Precision	F_1	Accuracy	Fallout
Worthäufigkeit	83.5 ± 5.6	79.2 ± 6.7	81.2 ± 5.8	95.1 ± 1.7	2.3 ± 0.5
Tf-Idf	83.5 ± 5.6	79.0 ± 6.7	81.1 ± 5.8	95.1 ± 1.7	2.5 ± 0.3
G^2	73.3 ± 5.9	71.5 ± 6.2	72.3 ± 5.8	92.9 ± 1.8	3.1 ± 0.6
Info Gain	70.5 ± 7.6	69.3 ± 5.2	69.6 ± 5.1	92.3 ± 1.6	3.6 ± 0.3
SVM-Gewichte	52.9 ± 7.0	49.2 ± 4.0	50.7 ± 4.0	87.2 ± 1.6	9.7 ± 0.7
SVM & Tf-Idf	79.9 ± 5.8	71.2 ± 5.3	75.1 ± 4.3	93.4 ± 1.6	5.1 ± 0.8

Tabelle 6.5.: Ergebnisse des gezielten Satzfilterns mit den verschiedenen ungekürzten Stichwortlisten, bei Anwendung der Stammformenreduktion.

menge verwendet wird. Der Recall-Precision-Breakeven-Punkt liegt kurz hinter diesem Schwellwert.

Dieses erste Experiment zum Satzfiltern verwendete die folgenden Parametereinstellungen (vgl. Seite 58): Erstellung der Ranglisten aus der satzweisen Markierung, Verwendung von Stammformenreduktion, Normierung des Satzgewichtes nach der Satzlänge, Auswahl der Sätze über das Satzgewicht und Verwendung der Wortgewichte aus der Rangliste. Außerdem wurde die Liste nicht, wie zur Kürzung von Sätzen, abgeschnitten. Die Resultate auf der Testmenge mit allen Listen zeigt Tabelle 6.5.

Aus dieser Tabelle wird deutlich, dass das Satzfiltern mit Themenwortliste der direkten Satzklassifikation, zumindest mit den verwendeten Textklassifikationsverfahren, überlegen ist. Nach dem F_1-Maß beträgt die Steigerung der besten Resultate gegenüber der Satzklassifikation mit Zentroidvektoren zwölf Prozentpunkte. Insbesondere der Precisionwert ist deutlich erhöht.

Es zeigen sich teilweise deutliche Unterschiede zwischen den verschiedenen Verfahren, eine Wortrangliste zu bestimmen. Worthäufigkeit und Tf-Idf-Gewicht sind im Ergebnis gleichwertig, während die Verwendung der SVM-Gewichte ohne zusätzliche Auswahl der Wörter nicht zu guten Werten führt (siehe jedoch Seite 69). Kombiniert man dagegen die SVM-Gewichte mit den Tf-Idf-Gewichten, wie in Abschnitt 4.2.5 erläutert, so sind die Ergebnisse gut. Dabei ist zu beachten, dass durch die Auswahl mit Tf-Idf nur etwas mehr als 100 Wörter übrig bleiben, die entstehende Rangliste also recht kurz ist. Die Methoden G^2 und Information Gain liefern ebenfalls zufriedenstellende Ergebnisse.

Es kann festgehalten werden, dass die vergleichsweise einfache Berechnung der Wortgewichte nach ihrer Häufigkeit oder ihrem Tf-Idf-Gewicht (mit Formel (4.1) auf Seite 34) zu signifikant besseren Resultaten geführt hat als die Verwendung komplizierterer und vor allem weniger effizienter Methoden. Damit ist der Aufwand für die Berechnung einer Stichwortliste sehr gering, wenn eine markierte Trainingsmenge vorliegt. Zum Aufwand folgen unten weitere Betrachtungen (Seite 71).

Mit der Methode der Auszählung der Worthäufigkeiten wurden mehr als 83 Prozent Recall bei 79 Prozent Precision erzielt. Demnach würde ein Benutzer des Filtersystems etwa jeden fünften terminbezogenen Satz verpassen, und von den ihm gezeigten Sätzen hätte gut jeder fünfte keinen Terminbezug. Der Accuracy-Wert zeigt, dass knapp fünf Prozent aller Sätze falsch klassifiziert werden.

Schwellwert	Recall	Precision	F_1	Accuracy	Fallout
0.04	82.5 ± 4.4	80.6 ± 5.8	81.4 ± 4.3	95.2 ± 1.4	2.5 ± 0.3
0.03	89.0 ± 4.3	74.0 ± 6.6	80.7 ± 4.6	94.6 ± 1.4	3.4 ± 0.4
0.02	95.0 ± 2.1	61.3 ± 6.7	74.3 ± 4.9	91.7 ± 1.7	6.5 ± 0.5
0.01	98.8 ± 1.6	35.6 ± 5.6	52.1 ± 6.2	77.1 ± 2.9	21.6 ± 1.0

Tabelle 6.6.: Ergebnisse des gezielten Satzfilterns mit der ungekürzten Worthäufigkeitsliste bei verschiedenen Schwellwerten.

Sehr erfreulich ist der durchgehend niedrige Fallout-Wert, der zeigt, dass nur zwei bis fünf Prozent aller nichtmarkierten Sätze irrtümlich für terminbezogen gehalten werden. Daher würde ein Benutzer des Systems nur recht wenige irrelevante Informationen zugeleitet bekommen. Die Zahl der Texte, die er lesen müsste, um die gewünschte Information zu finden, wäre also stark reduziert.

Durch Veränderung des Schwellwertes kann der Recallwert erhöht werden, etwa um weniger Information zu verpassen, wobei allerdings die Precision sinken wird. Die Frage ist, wie schnell der Precisionwert sinkt, wieviel man also für einen erhöhten Recall opfern muss. Tabelle 6.6 zeigt die Resultate auf den Testmengen bei künstlichem Absenken des Schwellwertes für die Worthäufigkeitsliste. In der ersten Zeile beträgt der Schwellwert 0.04. Dies entspricht dem Durchschnittswert für diese Liste aus Tabelle 6.5; dort hat der Schwellwert in den meisten der zehn Kreuzvalidierungsrunden diesen Wert und weicht in einigen davon ab. Hier dagegen wurde er in allen zehn Runden auf 0.04 gesetzt. In den folgenden Zeilen ist der Schwellwert jeweils um 0.01 gesenkt. Neben den Precisionwerten sind dabei auch die Falloutwerte von Interesse; zusammen zeigen beide Werte, dass die unnötig zu lesenden Sätze zwar zunehmen, aber immer noch eine deutliche Reduktion gegenüber der ungefilterten Textsammlung geleistet wird, wenn der Recall auf über 90 Prozent geschraubt wird.

Recall und Precision zu messen, macht jedoch nicht nur satzweise Sinn, sondern auch textweise, um zu beurteilen, wie viele terminbezogene Nachrichten *vollständig* verpasst werden bzw. umsonst gelesen werden müssen. Eine Nachricht, deren wichtigster terminbezogener Satz nicht erkannt wird, mag in anderen Sätzen dennoch Hinweise auf den Inhalt geben. Bei diesem Experiment gilt also jede markierte Email, aus der irgendein Satz über dem Schwellwert liegt, als erkannt. Diese textweise Messung von Recall und Precision ist also ein weniger strenges Maß, dementsprechend sind die Werte in Tabelle 6.7 höher als beim Satzfiltern selbst. Demnach wird ein Benutzer weniger als jede zehnte terminbezogene Nachricht verpassen, wenngleich er von einigen Nachrichten nicht die terminbezogenen Teile erhalten mag. Die Klassifikation der ganzen Emails auf diese Weise liefert übrigens bessere Resultate als die per SVM oder Zentroidvektoren, wie der Vergleich mit Tabelle 6.2 zeigt.

Im weiteren werden verschiedene Varianten des Verfahrens getestet. Tabelle 6.8 zeigt die Ergebnisse mit den gleichen Einstellungen wie bei Tabelle 6.5, aber ohne Stammformenreduktion. Daraus geht hervor, dass die Stammformenreduktion die Zusammenstellung der Themenwörter deutlich erleichtert. Tatsächlich zeitigte die Reduktion bei

Listenerstellung	Recall	Precision	F_1	Accuracy
Worthäufigkeit	93.1 ± 3.9	84.0 ± 5.9	88.2 ± 4.0	88.2 ± 3.9
Tf-Idf	93.1 ± 3.9	84.0 ± 5.9	88.2 ± 4.0	88.2 ± 3.9
G^2	91.2 ± 5.2	78.3 ± 5.8	84.2 ± 4.7	83.7 ± 4.6
Info Gain	89.4 ± 6.2	77.3 ± 5.5	82.8 ± 4.3	82.4 ± 3.0
SVM-Gewichte	76.6 ± 8.7	63.2 ± 6.9	68.8 ± 5.2	66.7 ± 6.3
SVM & Tf-Idf	93.4 ± 5.7	77.9 ± 5.9	84.8 ± 4.5	84.0 ± 4.5

Tabelle 6.7.: Ergebnisse des gezielten Satzfilterns aus Tabelle 6.5, emailweise gemessen.

Listenerstellung	Recall	Precision	F_1	Accuracy	Fallout
Worthäufigkeit	51.6 ± 10.2	40.7 ± 6.0	45.2 ± 6.7	83.4 ± 1.3	8.2 ± 0.8
Tf-Idf	51.3 ± 10.5	40.9 ± 5.9	45.1 ± 6.6	83.5 ± 1.4	8.5 ± 0.8
G^2	41.6 ± 7.0	32.2 ± 6.3	36.1 ± 6.1	80.2 ± 2.4	13.5 ± 1.3
Info Gain	65.5 ± 6.2	32.6 ± 4.5	43.3 ± 4.6	76.8 ± 3.0	20.7 ± 1.0
SVM-Gewichte	33.1 ± 9.3	17.5 ± 3.9	22.6 ± 5.3	70.2 ± 3.2	35.1 ± 1.4
SVM & Tf-Idf	41.4 ± 12.1	28.8 ± 4.0	33.1 ± 4.9	78.1 ± 2.4	15.6 ± 1.4

Tabelle 6.8.: Ergebnisse des gezielten Satzfilterns mit den verschiedenen ungekürzten Stichwortlisten, ohne Anwendung der Stammformenreduktion.

allen Experimenten durchgehend bessere Ergebnisse. Ohne Stammformenreduktion dagegen sind die Resultate kaum brauchbar. Die folgenden Experimente dieses Kapitels verwenden daher ausnahmslos Stammformen in den Wortlisten.

Wie im vorigen Kapitel erläutert ist, ist es sinnvoll, für die Satzkürzung die Wortranglisten in der Länge zu begrenzen, um hauptsächlich Wörter, die tatsächlich themenbezogen sind und die im oberen Teil der Liste erwartet werden, als semantische Hinweise zum Kürzen zu verwenden. Dafür wird die Liste beispielsweise nach 15 oder 20 Prozent ihrer Länge abgeschnitten. Die Frage liegt nahe, ob ein Abschneiden der Liste auch zur Satzextraktion sinnvoll ist. Es wurde bereits erläutert, dass die automatische Bestimmung des Schwellwertes die unterschiedlichen Längen und Gewichtsverteilungen der Listen ausgleichen sollte (Abschnitt 4.3). Dies wird im nächsten Experiment überprüft, dessen Ergebnisse Tabelle 6.9 angibt. Der Übersichtlichkeit halber ist in dieser Tabelle nur der F_1-Wert für verschiedene Listenlängen angegeben. Zum Beispiel bedeutet eine Listenlänge von 10% bei einer ursprünglichen Liste von 833 Wörtern, dass nur die ersten 83 Wörter zur Satzgewichtung herangezogen werden.

Wie man sieht, sind die Werte für verschiedene Längen tatsächlich ähnlich; unter Berücksichtigung der jeweils angegebenen Standardabweichung weichen sie nicht statistisch signifikant voneinander ab, auch nicht vom Ergebnis mit ungekürzten Listen. Der Grund dafür ist die erwartete Anpassung des Schwellwertes: Dieser wurde beispielsweise für die Worthäufigkeits-Liste bei der Listenlänge von 2% in den meisten Kreuzvalidierungsrunden zu 0.02 bestimmt, bei der Listenlänge von 40% dagegen zu 0.05. Da bei den längeren

	F_1-Wert bei Listenlänge				
Listenerstellung	2%	5%	10%	20%	40%
Worthäufigkeit	79.3 ± 5.0	78.4 ± 6.4	77.7 ± 5.4	77.1 ± 3.9	78.0 ± 4.4
Tf-Idf	79.2 ± 4.9	78.3 ± 6.7	77.6 ± 5.4	77.2 ± 4.0	78.1 ± 4.3
G^2	73.1 ± 5.3	71.8 ± 6.6	72.1 ± 5.5	72.1 ± 5.3	72.6 ± 5.1
Info Gain	70.6 ± 5.3	68.9 ± 6.2	67.7 ± 4.5	70.2 ± 4.6	70.6 ± 4.9
SVM-Gewichte	72.8 ± 6.4	71.4 ± 6.2	68.4 ± 4.4	65.6 ± 3.8	64.3 ± 5.0
SVM & Tf-Idf	59.4 ± 12.7	70.2 ± 4.4	74.4 ± 5.9	73.6 ± 4.0	73.1 ± 4.6

Tabelle 6.9.: Ergebnisse des gezielten Satzfilterns mit unterschiedlich gekürzten Stichwortlisten.

Listen mehr Wörter ein Gewicht haben, steigt das durchschnittliche Satzgewicht, was durch den höheren Schwellwert wieder ausgeglichen wird.

Eine Ausnahme zu der in Tabelle 6.9 erkennbaren, im wesentlichen geltenden Unabhängigkeit von der Listenlänge durch Anpassung des Schwellwertes bildet die per SVM erstellte Wortliste. Der Grund ist die sehr unterschiedliche Gewichtsverteilung der Wortgewichte in dieser Liste gegenüber den anderen Listen. Abbildung 6.4 zeigt, dass in den SVM-Listen im Vergleich zu den Tf-Idf-Listen die Wortgewichte auch weiter unten in der Liste noch sehr hoch sind. Die SVM unterscheidet also nicht so stark zwischen den Wörtern und gibt allen einen eher vergleichbaren Einfluss auf die Lage der trennenden Hyperebene (siehe Abschnitt 4.2.5). Dagegen sinken die Wortgewichte in den anderen Listen, deren Gewichtsverteilung der der Tf-Idf-Listen ähnelt, sehr schnell sehr stark ab. Dies führt dazu, dass die Wörter mit niedrigem Rang kaum Einfluss auf die Satzauswahl haben, während es bei den SVM-Listen umso schwieriger wird, den Einfluss der unteren Wörter gering zu halten, je größer der verwendete Teil der Liste ist. Bei Abschneiden der Liste nach einem kleinen oberen Teil liefert jedoch auch das Verfahren mit SVM-Gewichten zufriedenstellende Ergebnisse.

Interessanterweise lassen sich also gute Klassifikationsresultate bereits mit wenigen Themenwörtern erzielen. Bei 2% der ursprünglichen Listenlänge bleiben bei den Methoden, die nur den Wörtern im markierten Teil der Trainingstexte überhaupt ein Gewicht geben, 16–17 Stichwörter übrig (bei einem Testmengenanteil von durchschnittlich 0.1 der verwendeten Emailsammlung). Dies betrifft die Methoden der Worthäufigkeit, Tf-Idf und G^2 (vgl. Abschnitt 4.2). Bei der SVM-Methode mit Wortselektion nach Tf-Idf bleiben zunächst rund 100 Wörter übrig. Um so erstaunlicher ist das recht gute Ergebnis bei 5% Listenlänge (70.2% F_1), die Satzgewichtung erfolgt hier mit fünf oder sechs Themenwörtern. Dies sind Zeitangaben und das Wort `Treffen`. Demnach ist die Zusammensetzung der Emailsammlung so, dass diese wenigen Hinweiswörter für eine Performanz von 70 Prozent genügen. An dieser Stelle sei auf die Auswirkung der Zusammensetzung des Korpus auf den Precisionwert verwiesen, die in Abschnitt 4.3 auf Seite 38 erläutert wird. Demnach würde ein höherer (also auch realistischerer) Anteil von nicht terminbezogenen Emails die Precision-Ergebnisse senken und damit auch den F_1-Wert, ein Benutzer würde also mehr irrelevante Sätze zu sehen bekommen. Die Falloutwerte der bisherigen

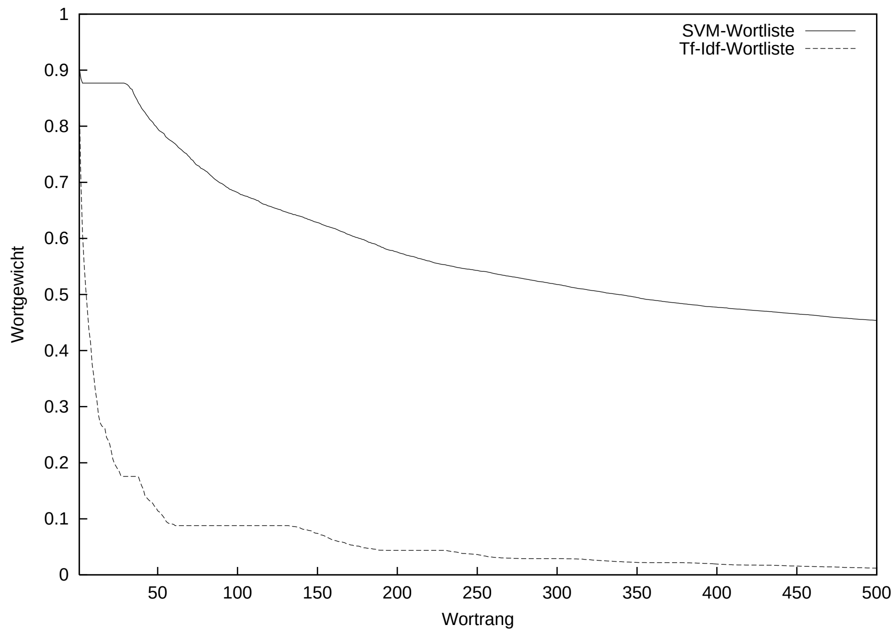

Abbildung 6.4.: Darstellung der Wortgewichtswerte der ersten 500 Wörter der SVM- und der Tf-Idf-Wortlisten (Durchschnitt von je zehn einzelnen Listen). Das erste Wort jeder Liste hat (wegen der Skalierung) immer das Gewicht 1.

Tabellen zeigen jedoch, dass dies in einem vertretbaren Maß geschieht.

Für die Satzkürzung (Kapitel 5) kann im übrigen eine andere Listenlänge verwendet werden als für die Satzauswahl. Dazu folgt in Abschnitt 6.3 mehr.

Es folgen Experimente zur Überprüfung der bisherigen Parametereinstellungen. Die Parameter sind auf Seite 58 aufgelistet. Sie können in beliebiger Kombination verwendet werden; hier sollen jedoch nicht alle Möglichkeiten der Kombination getestet werden. Statt dessen werden die folgenden Experimente zeigen, dass die Änderung der bisherigen Einstellung genau eines der Parameter jeweils zu keinem verbesserten Resultat führt. Durch weitere, hier nicht erwähnenswerte Experimente wurde auch sichergestellt, dass die gleichzeitige Änderung mehrerer Parameter die Resultate nicht verbessert.

Das Weglassen der Stammformenreduktion wurde bereits getestet (Tabelle 6.8 auf Seite 68). Der interessanteste der anderen Parameter ist sicherlich die Erstellung der Stichwortlisten. In den bisherigen Experimenten erfolgte diese durch Auszählen der Wortvorkommen etc. in den markierten Sätzen der Trainingsmenge, im Vergleich zu den nicht markierten Sätzen. Dazu ist die aufwändige satzweise Markierung der Textsammlung Voraussetzung. Wie die Abschnitte 4.1 und 4.2 erläutern, können die Verfahren zur Erstellung der Stichwortlisten jedoch unverändert übernommen werden, wenn nur eine textweise Markierung vorliegt. Es genügt dafür, jede Email als ganzes als terminbezogen oder nicht zu kennzeichnen, ein wesentlich niedrigerer Aufwand als die Markierung jedes terminbezogenen Satzes.

Listenerstellung	Recall	Precision	F_1	Accuracy	Fallout
Worthäufigkeit	82.9 ± 6.8	71.7 ± 6.6	76.5 ± 3.9	93.7 ± 1.1	3.4 ± 0.5
Tf-Idf	83.4 ± 7.3	71.7 ± 6.8	76.7 ± 4.3	93.8 ± 1.2	3.5 ± 0.6
G^2	74.9 ± 6.2	76.8 ± 8.5	75.7 ± 6.5	93.0 ± 1.6	4.7 ± 0.7
Info Gain	80.4 ± 8.1	71.8 ± 4.3	75.6 ± 4.7	93.6 ± 1.3	4.6 ± 0.5
SVM-Gewichte	84.9 ± 5.8	51.5 ± 5.3	63.9 ± 4.9	88.2 ± 1.6	10.3 ± 0.9
SVM & Tf-Idf	57.5 ± 6.0	85.5 ± 5.3	68.5 ± 4.4	93.5 ± 0.7	1.1 ± 0.3

Tabelle 6.10.: Ergebnisse des gezielten Satzfilterns bei Verwendung der oberen 2 Prozent von *textweise* berechneten Stichwortlisten.

In textweise berechneten Stichwortlisten kann erwartet werden, dass die Wortgewichtung weniger adäquat und genau erfolgt als in den satzweise erstellten, weil die positiv markierten Texte auch Sätze enthalten, die nicht markiert worden sind. Der markierte Teil der Sammlung enthält also mehr Wörter, die nicht zum Thema gehören, als bei der satzweisen Markierung. Da die bisherigen Ergebnisse aber gezeigt haben, dass die Satzextraktion auch mit wenigen Themenwörtern auskommt, lassen sich zufriedenstellende Resultate erwarten. Tabelle 6.10 gibt die Ergebnisse des Satzfilterns mit textweise erstellten Ranglisten an, aus denen nur die oberen zwei Prozent der Wörter verwendet wurden. Aufgrund der veränderten Wortgewichte verändern sich auch die jeweiligen Schwellwerte bei diesem Verfahren gegenüber der Verwendung satzweise berechneter Listen. Mit längeren Listenanteilen sinken die Ergebniswerte leicht, wegen der geringeren Adäquatheit der Listen.

Aus der Tabelle ist ersichtlich, dass auch die textweise Markierung der Trainingsmenge zu guten Resultaten führen kann. Dies ist ein sehr wichtiges Ergebnis, das den Arbeitsaufwand für die Vorbereitung des gezielten Satzfilterns stark reduzieren kann. Zumindest für die Termindomäne genügt es demnach, eine Beispielsammlung von Texten zu erstellen und jeden Text als themenbezogen oder nicht zu kennzeichnen. Mit einer satzweisen Markierung lässt sich die Genauigkeit der Extraktion steigern, aber nach diesen Ergebnissen nur um vier bis fünf Prozentpunkte gemessen am F_1-Wert. Ob dieser Aufwand gerechtfertigt ist, kann im Einzelfall je nach Anwendung entschieden werden. Dieses Resultat muss noch auf der anderen Textsammlung überprüft werden (Abschnitt 6.2.4).

Die folgende Tabelle 6.11 enthält wieder der Übersichtlichkeit halber nur die F_1-Werte. In der ersten Spalte sind die Werte aus Tabelle 6.5 wiederholt, die folgenden Spalten geben jeweils die Ergebnisse bei Veränderung genau eines der drei weiteren Parameter von Seite 58 an (die ersten beiden Parametereinstellungen wurden bereits getestet, siehe die Tabellen 6.8 und 6.10). Allen Werten außer denen der letzten Spalte liegen ungekürzte Wortranglisten zugrunde. Für die zweite Spalte wurde die Normierung des Satzgewichtes nicht wie bisher nach der Anzahl aller Wörter des Satzes vorgenommen, sondern nach der Anzahl der Wörter mit positivem Gewicht. Dadurch sollte es längeren Sätzen, deren themenbezogene Information nur in einem Satzteil steckt, leichter fallen, den Schwellwert zu übertreffen. Das Satzfiltern verbessert sich jedoch bei den meisten Verfahren nicht,

	F_1-Wert			
Listenerstellung	Standard	andere Satz-gewichtsnormierung	Auswahl über Phrasengewicht	Ignorieren der Wortgewichte
Worthäufigkeit	81.2 ± 5.8	75.2 ± 6.4	77.2 ± 4.3	76.7 ± 5.0
Tf-Idf	81.1 ± 5.8	75.3 ± 6.4	77.2 ± 4.3	76.8 ± 4.9
G^2	72.3 ± 5.8	64.1 ± 6.7	71.1 ± 4.2	64.8 ± 5.7
Info Gain	69.6 ± 5.1	69.6 ± 5.2	68.6 ± 4.0	56.0 ± 5.6
SVM-Gewichte	50.7 ± 4.0	71.7 ± 8.0	60.1 ± 6.0	69.3 ± 5.6
SVM & Tf-Idf	75.1 ± 4.3	72.3 ± 5.8	67.3 ± 7.0	75.4 ± 5.6

Tabelle 6.11.: Ergebnisse des gezielten Satzfilterns bei verschiedenen Einstellungen der Parameter von Seite 58. Die erste Spalte wiederholt die Ergebnisse aus Tabelle 6.5. Die weiteren Spalten werden im Text erläutert.

nur bei der Verwendung der SVM-Gewichte zeigt sich eine deutliche Verbesserung. Der Grund ist wieder in der unterschiedlichen Gewichtsverteilung (Abbildung 6.4) zu finden. Die hier verwendete Art der Normierung berechnet das Durchschnittsgewicht der Wörter mit positivem Gewicht eines Satzes. Dadurch lässt sich bei der SVM-Liste besser zwischen stark und niedrig gewichteten Wortgruppen unterscheiden als zuvor. Diese Unterscheidung fällt bei den anderen Listen wegen der größeren Unterschiede in den Wortgewichten auch bei der anderen Normierungsart leicht. Für die SVM-Methode empfiehlt sich daher diese Normierung, die bei den anderen Listen jedoch für ein wenig schwächere Resultate sorgt.

Die dritte Spalte gibt die Resultate an, die durch die Auswahl eines Satzes über das Gewicht einzelner seiner Phrasen entstehen. Statt des gesamten Satzgewichtes wird hier das Gewicht jeder einzelnen Phrase mit dem Schwellwert verglichen, und ein Satz wird dann ausgewählt, wenn mindestens eines seiner Phrasengewichte hoch genug ist. Dafür wurde die Phrasenerkennung von WAP benutzt. Dies führt zur Auswahl eines höheren Schwellwertes (zum Beispiel 0.1 bei der Worthäufigkeitsliste), weil die Phrasengewichte nicht durch die Anzahl aller Satzwörter geteilt werden. Ähnlich wie bei der vorigen Spalte ist die hinter diesem Vorgehen stehende Idee die Suche nach Satz*teilen*, die die gewünschte Information enthalten und deren Gewicht nicht durch die Länge eines Satzes mit irrelevanten Teilen heruntergesetzt werden soll (was wegen der Normierung geschehen würde). Obwohl die meisten automatisch erkannten Phrasen sich über weniger als einen Satzteil erstrecken, ergeben sich gute Resultate, die allerdings nicht besser sind als die bisherigen. Nach den Werten aus diesen beiden Spalten scheint die Konzentration auf Satzteile statt Sätzen also keine Verbesserung zu erbringen.

Für die letzte Spalte schließlich wurde die Idee untersucht, auf die Gewichtung der Wörter zu verzichten (Abschnitt 4.3) und die Themenwörter als ungeordnete Menge zu verwenden. Dieses Vorgehen ähnelt ein bisschen der einfachen Suche nach Stichworten, wie sie Suchmaschinen üblicherweise durchführen. Hierzu müssen die Wortlisten nach einem oberen Teil abgeschnitten werden, um wirklich hauptsächlich Themenwörter zu verwenden. Bei diesem Experiment wurden die ersten 5 Prozent der Wörter der Listen verwendet. Sie erhalten das Gewicht 1 zur Unterscheidung von allen anderen Wörtern. Hier würde übrigens die Normierung nach Anzahl der Gewichtswörter keinen Sinn mehr machen. Mit Normierung nach Anzahl der Wörter im Satz funktioniert aber die Schwellwertbestimmung wie zuvor. Die Ergebnisse deuten darauf hin, dass die Gewichtung der Wörter bei einigen Listen wichtiger ist als bei anderen, wie auch schon aus den Ausführungen zu Abbildung 6.4 hervorgeht. Bei den meisten Listen gelangen genug tatsächliche Themenwörter unter die ersten 5 Prozent, um die Satzauswahl auch ohne ihre Gewichtung vornehmen zu können. Bei Verwendung des Information Gain-Kriteriums ist die Gewichtung dagegen notwendig. Bei diesen Listen fällt das Wortgewicht noch stärker mit sinkendem Wortrang als bei den anderen und sorgt so für eine gute Unterscheidbarkeit der wichtigen von den unwichtigen Wörtern, auf die anscheinend nicht verzichtet werden kann.

Insgesamt erweist sich die Methode des Satzfilterns mit Schwellwertbestimmung als recht robust gegenüber verschiedenen Parametereinstellungen. Demnach gelingt es meist, den Schwellwert je nach den Gegebenheiten so auszuwählen, dass die Balance zwischen Auswahl zu vieler überflüssiger Sätze und Nichtauswahl relevanter Sätze gehalten wird. Die zunächst vorgestellten Einstellungen (wie zur Tabelle 6.5) sind leicht überlegen, daher werden sie für die weiteren Abschnitte dieses Kapitels als *Standardeinstellung* übernommen, soweit nicht anders vermerkt. Tabelle 6.12 gibt eine Übersicht.

Als weiteres wichtiges Experiment zur Beurteilung der Methoden dieser Arbeit folgt die Untersuchung ihrer Abhängigkeit von der Größe der Trainingsmenge. Wie in Abschnitt 6.2.1 erläutert, dient dies der weiteren Beurteilung des nötigen Aufwandes für die Vorbereitung des gezielten Satzfilterns. Tabelle 6.13 enthält die Ergebnisse bei den Standardeinstellungen unter Verwendung von etwa 10, 20, 50, 70 und 90 Prozent der Texte aus der Emailsammlung als Trainingsmenge. Die bisherigen Experimente verwendeten durchschnittlich 90 Prozent, die Werte der letzten Spalte stammen aus Tabelle 6.5.

Weiter oben wird gezeigt, dass die Terminvereinbarungen der verwendeten Emailsammlung bereits mit recht wenigen Stichwörtern erkannt werden können (die meisten davon beziehen sich auf Zeitangaben). Dementsprechend genügt eine vergleichsweise kleine Sammlung von Beispieltexten, um diese Stichwörter zu finden. So erklären sich die Resultate in Tabelle 6.13. Bei Verwendung von 20 Prozent der Texte dieser Sammlung zur Berechnung der Listen, also im Durchschnitt 112 Texten, lassen sich bereits sehr gute Resultate auf der dann recht großen Testmenge erzielen. Selbst bei Verwendung *textweiser* Markierung und 20 Prozent der Texte erzielt die Worthäufigkeitsliste nach einem gesonderten Experiment noch knapp 70 Prozent F_1. Aus dieser relativen Unabhängig-

Tabelle 6.12.: Übersicht über die Standardeinstellungen der Parameter von Seite 58.

Parameter	Einstellung	Parameter	Einstellung
Stammformenreduktion	ja	Normierung Satzgewicht	über Wortzahl
Listenberechnung	satzweise	Satzauswahl	über Satzgewicht
Listenlänge	100%	Wortgewicht verwenden	ja

	F_1-Wert bei Trainingsmengengröße				
Listenerstellung	10%	20%	50%	70%	90%
Worthäufigkeit	72.2 ± 3.3	76.4 ± 2.4	77.5 ± 1.6	78.9 ± 3.1	81.2 ± 5.8
Tf-Idf	72.4 ± 3.2	76.1 ± 2.5	77.5 ± 1.7	78.9 ± 3.0	81.1 ± 5.8
G^2	65.5 ± 4.5	70.3 ± 3.4	71.8 ± 2.8	73.1 ± 2.2	72.3 ± 5.8
Info Gain	66.9 ± 2.8	70.8 ± 2.3	70.3 ± 2.6	70.8 ± 2.6	69.6 ± 5.1
SVM-Gewichte	42.4 ± 5.3	47.8 ± 4.9	51.0 ± 3.6	51.0 ± 2.8	50.7 ± 4.0
SVM & Tf-Idf	63.0 ± 3.9	66.4 ± 5.3	71.5 ± 3.0	74.3 ± 1.9	75.1 ± 4.3

Tabelle 6.13.: Vergleich des gezielten Satzfilterns mit den Standardeinstellungen auf unterschiedlichen Anteilen der Emailsammlung als Trainingsmenge.

keit von der Trainingsmengengröße folgt auch, dass die eher geringe Größe der gesamten Emailsammlung die Aussagekraft der Resultate nicht einschränkt.

Für diese Emailsammlung kann also zunächst das Fazit gezogen werden, dass mit einfachen Mitteln (wenige Texte, textweise markiert) bereits eine Performanz von über 70 Prozent nach dem F_1-Maß erzielt werden kann, die durch die Verwendung einer größeren Trainingsmenge und gut abgestimmten Parametereinstellungen auf 80 Prozent erhöht werden kann, aber wohl nicht höher. Eine Erklärung dafür bieten die obigen Resultate, nach denen bereits wenige Themenwörter zu guten Ergebnissen führen können; diese Wörter sind also auch mit geringerem Aufwand zu finden. Warum sind aber 80 Prozent schwierig zu übertreffen? Dazu folgen Betrachtungen der Art von Sätzen, bei denen das Satzfiltern auf dieser Sammlung häufig Fehler macht.

Zwei Beispiele für Sätze, die markiert wurden, aber selten als terminbezogen erkannt werden, sind:

(1) `Da kann ich nicht.`

(2) `Am 14. ist Redaktion, da komme ich sowieso nach L und könnte dann doch besser dort vorbeischauen.`

Beispiel (1) zeigt eine Grenze des wortbasierten Satzfilterns auf, die in mehrfachen Wortbedeutungen begründet ist. Das Verb `können` wird in sehr vielen Zusammenhängen verwendet und kann daher nicht charakteristisch für Terminabsprachen sein, wird aber auch dort verwendet. Das Beispiel erhält seinen Terminbezug von den umgebenden Sätzen oder sogar von vorausgegangener Kommunikation (nicht notwendigerweise per Email) und ist daher mit dem Ansatz dieser Diplomarbeit nicht erfassbar.

Das zweite Beispiel ist ein längerer Satz, in dem eine Zeitangabe steckt, die aber aus den in den meisten Listen nicht sehr hoch gewichteten Wörtern `am` und `Number` (für die 14) besteht. Durch die Normierung nach der Satzlänge wird das Gewicht dieses Satzes zu klein. Bei der anderen Normierungsweise oder einem hohen Gewicht für `am` wird der Satz ausgewählt; dann entstehen aber andere Fehler mit anderen Sätzen (`am` tritt ja nicht nur bei Zeitangaben auf).

Das folgende ist ein Beispiel für fälschlicherweise extrahierte Sätze:

(3) `Das Foto wurde gegen 17:00 Uhr aufgenommen.`

Hier hat die Zeitangabe ausgereicht, das Satzgewicht über den Schwellwert zu bringen. Auch wenn „Foto aufnehmen“ ohnehin nicht als Termin zählt, verdeutlicht dieses Beispiel zusätzlich die Extraktion von Sätzen, die vergangene Daten behandeln, also keine Terminabsprachen sein können. Die Extraktion solcher Sätze ließe sich über die morphologische Analyse der Verben, die ja die Vergangenheitsinformation tragen, durch MESON in den meisten Fällen ausschließen; dies wäre aber ein domänenspezifisches Vorgehen, das bei der Suche nach anderen Informationen nicht angewandt werden kann. Immerhin könnte man solche Möglichkeiten den Benutzern optional anbieten. Im folgenden Beispiel ist das stets hoch gewichtete Wort `Termin` erwähnt und sorgt zusammen mit dem Wort `Zeit` oft für die Auswahl des Satzes, obwohl er nicht zu einer Terminabsprache gehört:

(4) `Mit Ruecksicht auf anschliessende Termine werden wir uns um ein straffes Programm bemuehen, um diese Zeit moeglichst einzuhalten.`

Es bleibt also festzuhalten, dass auch in einer der automatischen Erkennung relativ zugänglichen Domäne wie den Terminabsprachen so gut wie immer Fälle auftreten werden, die zu Fehlklassifikationen führen müssen. Daher ist eine Performanz von über 80 Prozent bereits sehr zufriedenstellend. Die Fehler liegen oft daran, dass Sätze im Allgemeinen so kurz sind, dass nur wenige inhaltliche Wörter darin auftauchen. Beim Satzfiltern hängt die Entscheidung, ob ein Satz über dem Schwellwert liegt, oft an einem einzigen Wort, was zu den beschriebenen unerwünschten Effekten führt. Die Extraktion über einen Schwellwert ist jedoch auch auf längere Texte anwendbar; dabei müssen dann zumeist mehrere Wörter zusammenwirken, um den Schwellwert zu überschreiten. Die höhere Redundanz ganzer Texte durch Mehrfachbenennung einiger Konzepte sollte sich also positiv auswirken, wie schon in Abschnitt 3.3.3 diskutiert wird. Tatsächlich übertrifft die Klassifikation von Emails mit Stichwortlisten, auch textweise erstellten, auf der Termindomäne die Werte der herkömmlichen Textklassifikation: beispielsweise erzielt die Worthäufigkeitsliste, textweise berechnet, 87.2 Prozent F_1 bei der Emailklassifikation. Hierbei ist das Vorgehen nicht dasselbe wie zur Tabelle 6.7: dort wird jede Email, aus der ein Satz extrahiert wird, als positiv klassifiziert verstanden, während hier die Email direkt über ihr Gesamtgewicht und einen Schwellwert klassifiziert wird. Das Ergebnis hier untermauert die obigen Erläuterungen und veranschaulicht den Vorteil längerer Texte gegenüber Sätzen.

Zum Abschluss der Analysen des direkten gezielten Satzfilterns auf dieser Domäne folgt noch ein Experiment zur Verwendung verschiedener Sprachstile. In Abschnitt 6.1.1 ist die Zusammensetzung der Emailsammlung aus privaten und geschäftlichen Emails beschrieben, wobei die privaten Emails im Sprachgebrauch etwas nachlässiger sind als die geschäftlichen, ebenso wie bei der Rechtschreibung. Um zu prüfen, ob sich dieser Unterschied auf das Satzfiltern auswirkt, können die Testmengen für private und geschäftliche Emails gesondert betrachtet werden. Tabelle 6.14 gibt die entsprechenden Ergebnisse bei der Standardeinstellung an. Wegen der Einschränkung auf private oder geschäftliche Emails werden die Testmengen zu klein für statistische Aussagen, wenn die Trainingsmenge 90 Prozent der Daten beinhaltet; diese Tabelle beruht daher auf dem Trainingsmengenanteil von 70 Prozent aus Tabelle 6.13. Es wurde nicht für jede Gruppe von Emails gesondert trainiert.

	F_1-Wert		
Listenerstellung	privat	geschäftlich	gemischt
Worthäufigkeit	72.5 ± 2.5	83.0 ± 3.8	78.9 ± 3.1
Tf-Idf	72.5 ± 2.5	82.9 ± 3.7	78.9 ± 3.0
G^2	71.2 ± 3.5	74.3 ± 4.2	73.1 ± 2.2
Info Gain	67.7 ± 4.6	72.8 ± 4.2	70.8 ± 2.6
SVM-Gewichte	41.5 ± 5.1	57.8 ± 4.9	51.0 ± 2.8
SVM & Tf-Idf	66.1 ± 4.0	79.6 ± 2.5	74.3 ± 1.9

Tabelle 6.14.: Vergleich des gezielten Satzfilterns mit den Standardeinstellungen auf privaten und geschäftlichen Emails; Verwendung von durchschnittlich 30 Prozent der Textsammlung für die Testmengen.

Listenerstellung	Recall	Precision	F_1	Accuracy	Fallout
Worthäufigkeit	73.3 ± 7.9	91.8 ± 4.2	81.2 ± 5.4	95.4 ± 1.4	1.1 ± 0.6
Tf-Idf	73.3 ± 7.9	91.8 ± 4.2	81.2 ± 5.4	95.4 ± 1.4	1.1 ± 0.6
G^2	64.1 ± 5.8	89.1 ± 5.9	74.3 ± 4.9	94.0 ± 1.3	1.3 ± 0.8
Info Gain	62.9 ± 6.4	86.2 ± 6.4	72.5 ± 5.4	93.5 ± 1.3	1.6 ± 0.9
SVM-Gewichte	54.7 ± 9.2	81.7 ± 3.2	65.0 ± 6.2	92.1 ± 1.5	2.0 ± 0.6
SVM & Tf-Idf	72.6 ± 7.5	85.9 ± 6.2	78.1 ± 2.9	94.5 ± 1.1	2.0 ± 1.1

Tabelle 6.15.: Ergebnisse des indirekten gezielten Satzfilterns bei Verwendung der Zentroidklassifikation für ganze Emails und Standardeinstellungen.

Nach diesen Zahlen zahlt sich ein sorgfältigerer Sprachgebrauch beim gezielten Satzfiltern aus. Ein auf den Methoden dieser Diplomarbeit beruhendes Filtersystem könnte also am erfolgreichsten im beruflichen Umfeld eingesetzt werden, solange der sprachliche Umgang dort nicht zu umgangssprachlich ist. Allerdings mag bei diesem Ergebnis auch eine Rolle spielen, dass man im Beruf genauere Zeitangaben verwenden wird als privat, wo es auch schon mal Verabredungen für „demnächst“ geben kann.

Indirektes gezieltes Satzfiltern

In diesem Abschnitt folgt ein kurzer Test zum in Abschnitt 4.3 motivierten indirekten Satzfiltern über vorherige Textklassifikation. Da die Zentroidklassifikation auf den Emails besser abschneidet als die SVM, wird sie hier verwendet. Alle von ihr *positiv* klassifizierten Emails (vgl. Tabelle 6.2) werden mit den Stichwortlisten nach themenbezogenen Sätzen gefiltert, aber keine anderen. Als Grundlage für die Bewertung dient jedoch wie bisher die Satzmenge aller jeweiligen Testemails, auch der negativ klassifizierten. Tabelle 6.15 gibt die Resultate. Dabei wurden für das Satzfiltern die Standardeinstellungen verwendet. Die Ranglisten wurden nicht neu berechnet, sondern die passenden Listen aus dem vorigen Abschnitt übernommen, um den Einsatz bei fertig vorliegender Liste zu testen. Dies macht Sinn, weil die Textklassifikation zur schnellen Vorauswahl der Emails dienen soll

Listenerstellung	Recall	Precision	F_1	Accuracy	Fallout
Worthäufigkeit	75.4 ± 6.5	69.6 ± 7.2	71.9 ± 4.0	93.8 ± 1.3	3.5 ± 0.2
Tf-Idf	75.3 ± 6.5	69.7 ± 7.1	71.9 ± 4.0	93.8 ± 1.3	3.4 ± 0.1
G^2	47.3 ± 7.7	39.1 ± 8.1	41.8 ± 4.7	86.3 ± 1.6	11.4 ± 0.2
Info Gain	48.2 ± 6.7	46.5 ± 9.2	46.8 ± 5.5	88.5 ± 1.7	6.3 ± 0.2
SVM-Gewichte	44.3 ± 5.8	47.2 ± 7.3	45.2 ± 3.9	88.7 ± 1.8	6.8 ± 0.2
SVM & Tf-Idf	48.7 ± 6.0	53.5 ± 7.7	50.6 ± 5.0	90.1 ± 1.4	5.7 ± 0.1

Tabelle 6.16.: Ergebnisse des gezielten Satzfilterns nach Wahlergebnissen, bei Standardeinstellungen und einer durchschnittlichen Testmengengröße von 20 Prozent aller Texte.

(Abschnitt 4.3) und danach eine Stichwortlistenberechnung nicht mehr erfolgen soll, um die Laufzeit gering zu halten. Ebenso wurde der Schwellwert übernommen, der mit den entsprechenden Ranglisten bereits ermittelt wurde. Das Szenario entspricht also einer Anwendung mit fertig trainiertem Klassifikator und vorhandenen Stichwortlisten mitsamt Schwellwert, so dass die Filterung sehr schnell geht.

Aus Tabelle 6.15 geht hervor, dass dieses Vorgehen gute Ergebnisse erbringen kann, die mit denen des direkten Satzfilterns vergleichbar sind. Die hohen Precisionwerte erklären sich durch den hohen Precisionwert der zentroidbasierten Textklassifikation in Tabelle 6.2. Die Qualität der Textklassifikation schlägt also durch, was nicht überraschen kann. Ob dieses Vorgehen Sinn macht, hängt von der Zusammensetzung der Daten ab und dem möglichen Aufwand zum Training der Textklassifikationsmethode sowie ihrer Qualität. Wenn ein guter, fertig trainierter Klassifikator vorliegt und nur wenige relevante Texte in den Daten erwartet werden, wird die Vorauswahl der Texte die Suche nach einzelnen Sätzen beschleunigen, weil nicht jedes Satzgewicht jedes irrelevanten Textes berechnet werden muss; dabei müssen keine Performanzeinbußen hingenommen werden.

6.2.4. Wahlergebnisse

In diesem Abschnitt geht es darum, das gezielte Satzfiltern, das im Mittelpunkt dieser Diplomarbeit steht, auf einer zweiten Domäne zu testen, um die bisherigen Aussagen über das Verfahren zu untermauern. Hier werden also Satzklassifikation mit Textklassifikationsverfahren sowie indirektes Satzfiltern nicht mehr weiter untersucht.

Für die Experimente mit den Wahlergebnissen wurden die Standardeinstellungen aus Tabelle 6.12 übernommen. Es soll geprüft werden, ob sich die Anwendung auf ein neues Thema wirklich schnell und einfach bewerkstelligen lässt, weshalb lange, erneute Tests zu den Parametereinstellungen nicht angebracht sind. Allerdings wird die Verwendung der textweise erstellten Ranglisten, neben den satzweise erstellten, ebenfalls untersucht, da dies den Aufwand stark senkt.

Tabelle 6.16 gibt die Resultate des Satzfilterns nach Wahlergebnissen an, mit satzweise erstellten Ranglisten. Der Anteil der Trainingsmenge an allen Texten beträgt 80 Prozent, damit die Testmengen nicht zu klein werden.

Die Ergebniswerte weisen starke Unterschiede auf, liegen jedoch für zwei Wortlisten über 70 Prozent. Gezieltes Satzfiltern ist also auch auf dieser Domäne erfolgreich. Die Unterschiede können wie folgt erklärt werden: In den Wortlisten mit schlechten Resultaten stehen die Wörter `Prozent` sowie `Number` als Stellvertreter für Zahlen ganz oben, auf den ersten beiden Plätzen. Die meisten Sätze, die eine Prozentangabe enthalten, müssen also extrahiert werden. Jedoch enthält die Nachrichtensammlung auch Texte aus dem Wirtschaftsressort und anderen Bereichen, die Prozentangaben enthalten; Zahlen sind ohnehin in allen Bereichen zu finden. Folglich werden mit diesen Listen viele Sätze fälschlich ausgewählt. Bei den beiden Listen, die zu guten Resultaten führen (Worthäufigkeit und Tf-Idf), steht das Wort `Prozent` hinter der 50. Stelle, `Number` kommt erst hinter der 300. Stelle. Bei diesen Wortlisten stehen oben andere Wörter aus dem Bereich „Wahlen"; in Anhang A.3 findet sich eine Beispielliste.

Auf dieser Textsammlung sind also die einfachen Verfahren noch deutlicher überlegen. Fehler entstehen dennoch bei Sätzen, die wirtschaftliche Entwicklungen mit Prozentangaben beschreiben, aber auch bei Hintergrunderläuterungen zu den Wahlen, wie die folgenden Beispiele verdeutlichen.

(5) `Im vergangenen Jahr legte bei Sixt das Vermietgeschäft nur um 1,3 Prozent zu.`

(6) `Viele Minderjährige - ohnehin nicht wahlberechtigt - sollen mehr als eine Stimme abgegeben haben.`

Diese Sätze wurden wegen der Wörter `zulegen`, das unter den ersten 30 Wörtern der Worthäufigkeitsliste zu finden ist, und `Prozent` bzw. `Stimme` irrtümlich extrahiert. Das folgende Beispiel ist ein Satz, der wohl wegen seiner indirekten Ausdrucksweise nicht ausgewählt wurde, obwohl er ein Wahlergebnis behandelt.

(7) `Bei den gleichzeitig stattfindenden Kommunalwahlen im bevölkerungsreichsten Bundesland Nordrhein-Westfalen erlitten die Sozialdemokraten flächendeckende Einbrüche und mußten traditionelle Hochburgen verloren geben.`

Es kann zu dieser Domäne gesagt werden, dass ihr Vokabular viele Überschneidungen mit wirtschaftlichen und sportlichen Nachrichten aufweist, die eine satzweise Unterscheidung anhand von Stichwörtern in einigen Fällen unmöglich machen. Dennoch wird mit den verwendeten Verfahren eine starke Reduktion der für Wahlergebnisse in Frage kommenden Textmenge erreicht. Für einen gewinnbringenden Einsatz wird es wohl notwendig sein, den Schwellwert für die Satzauswahl künstlich zu senken, um den Recallwert zu verbessern, also nicht so viele Wahlergebnisse zu verpassen. Ein entsprechendes Experiment führte zu 55 Prozent Precision und 7 Prozent Fallout bei 95 Prozent Recall, was immer noch einer starken Reduktion entspricht.

Ein wichtiges Experiment aus dem vorigen Abschnitt ist die Verwendung textweise berechneter Stichwortlisten, die dort zu nur leicht verschlechterten Ergebnissen geführt hat. Dies betrifft den notwendigen Aufwand für die Vorbereitung des Satzfilterns. Wie zuvor sollte hierbei die Stichwortliste gekürzt werden, um nur Wörter aus dem oberen Teil zu verwenden; zwar wird dies auch mit dem Schwellwert erzielt, aber die Kürzung ist zusätzlich wegen der geringeren Genauigkeit der Stichwortermittlung sinnvoll (bei den

Terminen sind die kürzeren Listen besser, wenn sie textweise ermittelt wurden; siehe Seite 71).

Leider können für diese Textsammlung die guten Resultate der Termindomäne mit textweise erstellten Wortranglisten nicht erzielt werden. Der Grund liegt in der Beschaffenheit der verwendeten Texte. Die einzelnen Nachrichtentexte sind wesentlich länger als die Emails und behandeln mehr Themen; etliche Artikel sind Übersichten über mehrere Nachrichten. Daher gelangen bei der textweisen Markierung deutlich mehr nicht themenbezogene Wörter in positiv markierte Texte. Den entstehenden Ranglisten ist ein deutlicher Unterschied anzumerken, je nach Entstehungsweise: bei textweise erstellten Listen stehen deutlich mehr allgemeinpolitische Wörter oben als bei den satzweise erstellten, zum Beispiel `liberal`, `Koalition`, `Partei` und ähnliche. Ein Auszug aus einer solchen Liste findet sich in Anhang A.3. Es bildet sich durch die textweise Markierung eine anderere Abgrenzung des Themas „Wahlen", nämlich zu Artikeln aus anderen Zeitungsressorts wie Wirtschaft usw. Dies änderte sich auch nicht bei einem Zusatzexperiment, bei dem die Nachrichten der Deutschen Welle, die immer Übersichten über verschiedene Ereignisse enthalten, entfernt wurden.

Trotzdem sind einige Wörter wie `Wahlsieg`, `erzielen` oder `Stimme`, die in den satzweise erstellten Listen oben stehen, auch in den textweise erstellten Listen oben zu finden (siehe Anhang A.3) und sorgen dafür, dass die Satzextraktion immer noch weit besser als der Zufall ist: das beste Ergebnis nach dem F_1-Maß, erzielt mit der textweise erstellten Worthäufigkeitsliste, beträgt 59.5 Prozent, errechnet aus immerhin 75 Prozent Recall und 50 Prozent Precision, bei 2.2 Prozent Fallout. Bei Benutzung dieser Einstellungen würde man also ein Viertel der Sätze mit Wahlresultaten verpassen, und nur jeder zweite zu lesende Satz enthielte tatsächlich Wahlergebnisse. Ohne jedes Filtern enthielte allerdings, bei dieser Nachrichtensammlung, nur jeder zehnte Satz Wahlergebnisse. Die Filterung präsentiert den Benutzern nur gut zwei Prozent aller Sätze, die nicht von Wahlergebnissen handeln.

Gezieltes Satzfiltern erweist sich nach der Auswertung dieses Kapitels als flexibles und erfolgreiches Verfahren, dessen Vorbereitungsaufwand von der gestellten Aufgabe, der Art der Texte und der zu erzielenden Qualität des Filterns abhängt. Sind die Texte vergleichsweise kurz und tendieren sie nicht dazu, viele verschiedene Themen zu behandeln, so genügt die textweise Markierung der Trainingsmenge, um eine Liste von Wörtern zu erhalten, die für das interessierende Thema charakteristisch sind und mit der erfolgreich relevante Textsegmente automatisch selektiert werden können. Dabei kann die Auswahl in Bezug auf Recall und Precision leicht gesteuert werden. Liegt die Liste einmal vor, so ist die weitere Extraktion schnell und effizient, soweit die notwendige Stammformenreduktion es ist.

6.3. Satzkürzung und SMS-Texte

Dieser Abschnitt analysiert die Methoden aus den Abschnitten 5.1 und 5.2 zur Satzkürzung und Erstellung von SMS-Texten. Hier liegt also wieder die Emailsammlung von Terminabsprachen zugrunde.

Zunächst sei daran erinnert, dass die Kürzung der Sätze entscheidend von den Stichwortlisten abhängt, deren Wörter bestimmen, welche Satzteile (Wörter oder Phrasen) ein

Gewicht erhalten und daher nicht gestrichen werden können. Da die Stichwortlisten nur im oberen Teil wirkliche Themenwörter enthalten, müssen sie nach einer gewissen Länge abgeschnitten werden. Angesichts der Verteilung von Terminwörtern in einer typischen Liste (Abbildungen 6.1 und 6.2 sowie Abschnitt A.2) ist nicht von vorneherein klar, an welcher Stelle dieses Abschneiden erfolgen sollte.

Die Experimente dieses Abschnitts werden sämtlich mit den Worthäufigkeitslisten durchgeführt, da sie beim Satzfiltern zusammen mit den Tf-Idf-Listen zu den besten Resultaten führen, aber einfacher zu errechnen sind als diese. In Abbildung 6.2 auf Seite 65 ist erkennbar, dass in solchen Listen Terminwörter auch um die 170. Stelle von 833 Wörtern auftreten. Danach werden sie allerdings selten. Für die Satzkürzung werden die Listen daher nach 20 Prozent ihrer Länge abgeschnitten. Dies bedeutet, dass auch viele nicht terminbezogene Wörter noch in der Liste verbleiben und für positives Gewicht von Satzteilen sorgen, von denen manche irrelevant sein werden. Jedoch würden die Kürzungen offensichtlich zu stark werden, wenn man nur 2 oder 5 Prozent der Liste verwenden würde, so dass 20 Prozent als guter Kompromiss erscheinen, der auch für bessere Lesbarkeit sorgen kann.

6.3.1. Statistiken

Zuerst wird untersucht, wie stark die einzelnen in Tabelle 5.1 auf Seite 45 aufgelisteten Kürzungsstufen sich auf Sätze der Sammlung auswirken. Die Untersuchung kann nur mit Sätzen geschehen, die beim Satzfiltern als terminbezogen ausgewählt wurden, da andere Sätze kein oder wenig Gewicht haben und daher zu stark gekürzt würden. Die Tabelle 6.17 bezieht sich deshalb auf extrahierte Sätze aus den Testmengen zum Hauptexperiment zum gezielten Satzfiltern, dessen Resultate in Tabelle 6.5 auf Seite 66 dargestellt sind.

Aus Tabelle 6.17 wird deutlich, dass die Durchschnittslänge der extrahierten Sätze 88 Zeichen beträgt, also über der in Abschnitt 6.1.1 erwähnten Durchschnittslänge aller Sätze von 78 Zeichen liegt. Die Tabelle stellt dann in jeder Zeile die Durchschnittslänge aller extrahierten Sätze nach Anwendung aller bis dahin angegebenen Kürzungsstufen dar. Für diese Tabelle wurde die Halbsatzentfernung (siehe Abschnitt 5.1, Seite 48) nicht durchgeführt, um die Kürzungsstufen einzeln zu betrachten; bei der SMS-Erstellung wurde sie jedoch stets eingesetzt, entsprechend dem Algorithmus in Abbildung 5.1. Neben der Durchschnittslänge in Zeichen ist angegeben, welchem Reduktionsfaktor dies entspricht, sowie der Anteil an allen extrahierten Sätzen, die bis zu dieser Stufe von Kürzungen betroffen sind. Die letzte Stufe erreicht also 93 Prozent der Sätze und verkürzt Sätze durchschnittlich auf etwas über die Hälfte. Die nie gekürzten Sätze sind sehr kurze Sätze mit Zeitangaben, die daher kein gewichtsloses Wort haben, wie das folgende Beispiel illustriert.

(8) `Beginn: Samstag, 12h. Ende: Sonntag, 14h.`

Diese Messungen sagen nichts über die Lesbarkeit der Sätze oder die Adäquatheit der Kürzungen aus. Hierzu folgen in Abschnitt 6.3.2 die Untersuchungen mit Testpersonen. Vorher werden noch einige Angaben zur automatischen SMS-Erstellung nach den Algorithmen aus Kapitel 5 gemacht.

Zum Beispiel ist interessant, dass in der Termindomäne durchschnittlich 68 Prozent der erstellten SMS-Nachrichten ohne Satzkürzung auskommen, das heißt das Satzfiltern

Stufe	Satzlänge	Reduktionsfaktor	Betroffene Sätze (in %)
0	88	1.0	-
1	85	0.96	40
2	80	0.90	51
3	78	0.88	71
4	73	0.83	84
5	67	0.76	89
6	64	0.72	90
7	64	0.72	90
8	62	0.70	91
9	59	0.67	91
10	59	0.67	91
11	56	0.63	91
12	46	0.52	93

Tabelle 6.17.: Wirkung der Kürzungsstufen auf als terminbezogen extrahierte Sätze (Durchschnitt von 465 extrahierten Sätzen aus den Testmengen von zehn Kreuzvalidierungsrunden). Die Nummern der Stufen beziehen sich auf Tabelle 5.1.

liefert hier Sätze, die kurz genug sind, um in 160 Zeichen wiedergegeben zu werden. Weiterhin genügt es bei 82 Prozent der SMS, Kürzungen nur bis zur Stufe sechs vorzunehmen, die noch als sehr sicher betrachtet werden kann in dem Sinne, dass nur selten wichtige Informationen gestrichen werden (Kapitel 5). Damit sollte ein Großteil der SMS, zumindest für diese Emailsammlung, gut lesbar sein. Bei den meisten der anderen SMS allerdings (über 13 Prozent) genügt dann auch Stufe zwölf nicht, um alle ausgewählten Sätze in der SMS unterzubringen. Hierbei sind meistens so viele Sätze des Originaltextes ausgewählt worden, dass eine Kürzung auf 160 Zeichen in lesbarer Form unmöglich ist.

Im Durchschnitt müssen die extrahierten Sätze bis zur Stufe fünf gekürzt zu werden, um eine SMS zu bilden; dabei sinkt ihre Durchschnittslänge—mit Halbsatzentfernung—von 88 auf 61 Zeichen. Kürzt man die Sätze nicht, so passen im Schnitt 83 Prozent der extrahierten Sätze einer Email in die SMS; bei Kürzung bis zur Stufe sechs sind es 89 Prozent und bei Maximalkürzung 93 Prozent. Dabei werden nicht vollständig untergebrachte Sätze anteilig nach Anzahl der untergekommenen Wörter mitgezählt.

Es werden einige SMS zu Emails erstellt, die keinen Terminbezug haben. Zu den anderen Emails gibt es aber keine SMS, die nicht wenigstens einen markierten Satz enthält. Zumindest wird also nicht völlig am interessanten Inhalt vorbei extrahiert. Dies schließt nicht aus, dass zu Emails mit Terminbezug keine SMS erstellt wird.

6.3.2. Informationsgehalt der SMS-Nachrichten

In Abschnitt 3.5.2 wird die Bewertung von Zusammenfassungen in intrinsische und extrinsische Bewertungen unterschieden. Intrinsische Bewertungen messen die Qualität der Zusammenfassungen. Eine solche Bewertung wird in diesem Abschnitt vorgenommen, indem der Informationsgehalt der Zusammenfassungen untersucht wird. Eine Anzahl von automatisch nach den Verfahren aus Kapitel 5 erstellten SMS-Nachrichten wurde dazu Testpersonen vorgelegt, die inhaltliche Fragen zu den vorkommenden Terminabsprachen beantworteten. Wie im erwähnten Abschnitt diskutiert, ist die inhaltliche Bewertung gezielter Zusammenfassungen leichter, weil bekannt ist, wonach gesucht wird. Dementsprechend können sich die Fragen hier auf Terminabsprachen konzentrieren und auf diese Weise testen, in welcher Weise die vorgestellten Verfahren ihren Zweck erfüllen. Der sonst so schwierig zu bemessende Informationsgehalt von Zusammenfassungen kann also hier vergleichsweise fundiert beurteilt werden.

Im Vordergrund steht bei dieser Beurteilung der Vergleich mit dem Informationsgehalt der Ursprungstexte. Die Befragung konzentriert sich nur auf Informationen, die aus dem Text ohne Kenntnis des Kontextes (etwa Absender, Empfänger, vorhergegangene Kommunikation, Hintergrundwissen) entnommen werden können. Es werden Abweichungen gemessen zwischen Angaben, die aufgrund der Lektüre der ganzen Emails gemacht wurden, und solchen, die nur bei Kenntnis einer SMS-Nachricht gemacht wurden. Auf diese Weise muss nicht beurteilt werden, wie richtig oder wie falsch eine Antwort in Bezug auf den Originaltext ist; da auch bei den Originaltexten der Kontext fehlt, ist dies oft nicht möglich. Es gibt also Fragen, die auch bei Kenntnis der ganzen Email nicht beantwortbar sind, außer für den Absender und eventuell den Empfänger. Dies ist zu berücksichtigen, wenn eine Antwort bei den Fragen zur SMS-Nachricht fehlt.

Für die Bewertung wurden Emails zufällig aus Testmengen ausgesucht, die an Kreuzvalidierungsrunden teilnahmen, deren Ergebnis nicht zu stark vom Durchschnitt der zehn Runden abwich, und für die mindestens ein Satz extrahiert wurde. Um auch die verschiedenen Kürzungsstufen untersuchen zu können, wurden zu jeder verwendeten Email maximal drei verschiedene SMS-Nachrichten gebildet, auf folgende Weise. Die erste SMS wird mit Sätzen, die maximal bis zur Stufe sechs gekürzt sind, erstellt (leichte Kürzung). Die zweite SMS verwendet Sätze, die maximal bis zur Stufe neun gekürzt sind (mittlere Kürzung). Wenn keine über Stufe sechs hinausgehende Kürzung notwendig oder wirksam ist, sind diese beiden SMS-Nachrichten gleich und nur die erste wird verwendet. Sonst wird die dritte SMS erstellt, sie darf maximal bis zur Stufe zwölf gekürzte Sätze enthalten (starke Kürzung). Sind keine über Stufe neun hinausgehenden Kürzungen notwendig oder wirksam, so ist die dritte Nachricht gleich der zweiten und nur die ersten beiden werden verwendet. Es gibt also zwischen zwei und vier Versionen jedes Textes, der an der Bewertung teilnimmt: Es gibt immer die Ursprungsemail und die leicht gekürzte SMS-Nachricht, dazu eventuell die mittel und stark gekürzten Nachrichten.

Da die starke Kürzung von Sätzen nur vergleichsweise selten nötig ist (siehe vorigen Abschnitt), wurde die Auswahl der Emails dahingehend beeinflusst, genügend SMS-Nachrichten mit starker Kürzung zu verwenden, um Aussagen über solche Nachrichten machen zu können. Die restlichen Mails wurden zufällig ausgewählt. Insgesamt sind dies 50 Emails, zu denen 50 SMS mit leichter Kürzung, 24 mit mittlerer und 23 mit starker Kürzung gehören, zusammen 147 Texte. Unter den 50 Mails sind 7 ohne Terminbezug, aus denen dennoch Sätze extrahiert wurden. Den Testpersonen wurden jeweils zwei oder drei Texte vorgelegt; keine Testperson sah zwei Versionen des gleichen Textes. Die etwas mehr als 50 Testpersonen sind zum großen Teil identisch mit vielen der Personen, die Emails für die Sammlung zur Verfügung stellten; jedoch wurde niemandem ein Text, der von ihm stammte, vorgelegt, so dass niemandem der Kontext seiner Texte bekannt war.

Den Testpersonen wurde erläutert, dass es um Terminabsprachen geht. Der Aufbau der automatisch erzeugten SMS-Nachrichten mit dem Informationsstring zu Beginn und den Sonderzeichen, die ausgelassene Teile bezeichnen, wurde ebenfalls erklärt (siehe Abschnitt 5.2). Die folgenden Fragen waren zu beantworten (zitiert aus dem Anschreiben an die Testpersonen):

1. Art des Treffens (zum Beispiel: Arbeitssitzung, Picknick, Vortrag, Probe, Spiel, Konferenz, Party ...)

2. Zeitpunkt, zu dem das Treffen stattfinden soll

3. Ort des Treffens

4. Beteiligte Personen und/oder Organisationen

5. „Status“ des Treffens - bitte ankreuzen bzw. eine Antwort auswählen:

    a) wird angekündigt
    b) wird vorgeschlagen
    c) fällt aus
    d) ist schon vorbei
    e) verschiebt sich
    f) jemand sagt zu
    g) jemand sagt ab
    h) nicht beantwortbar
    i) sonstiges (bitte kurz angeben)

6. Wie lesbar und verständlich findest Du den Text? Bitte eine Zahl von 1 (unverständlich) bis 5 (gut verständlich) angeben.

Dazu wurde erklärt, dass nicht immer alle Fragen beantwortet werden können und dass auch Texte dabei sein können, in denen kein Termin zu finden ist. Zur Beantwortung der Fragen durften alle Teile der Nachrichten inklusive Header der Emails herangezogen werden.

Zur letzten Frage nach der Lesbarkeit bzw. Verständlichkeit des Textes ist zu sagen, dass die Antworten natürlich sehr subjektiv sind. Beim Vergleich der Antworten fallen starke Unterschiede auf. Manche Testpersonen gaben an, dass sie glaubten, der Text sei gut verständlich für den ursprünglichen Empfänger, da diesem der Kontext bekannt sei; nach diesem Hinweis folgten jedoch beliebige Zahlen zwischen eins und fünf. Die Antworten zu dieser Frage können daher nur eine durchschnittliche Tendenz angeben, die allerdings die Erwartungen bestätigt (siehe unten).

Da es nicht um Korrektheit der Antworten geht, werden bei den Antworten zu den ersten fünf Fragen folgende Fälle unterschieden:

1. Die Frage ist weder bei Kenntnis der ganzen Email noch der SMS beantwortbar.

2. Die Frage ist für die Email beantwortbar, aber nicht für die SMS.
3. Die Frage ist für die Email nicht beantwortbar, wurde aber für die SMS beantwortet.
4. Die Antwort ist für die SMS die gleiche wie für die Email.
5. Die Antwort für die SMS ist kürzer oder weniger detailliert als für die Email.
6. Die Antwort für die SMS weicht von der für die Email ab.

Die Präsentation aller Prozentzahlen zu Übereinstimmung, Abweichung etc. bei den Antworten zu jeder der fünf Fragen, jeweils für die drei SMS-Varianten, wäre etwas unübersichtlich, so dass im folgenden nur die wesentlichen Resultate der Auswertung zusammengefasst werden.

- Einige der Emails, die keinen Terminbezug aufweisen, wurden bei Vorlage des Originaltextes als solche erkannt, jedoch bei Kenntnis nur der SMS als terminbezogen missverstanden. Bei anderen war dies insofern nicht der Fall, als für die SMS alle Fragen mit „nicht beantwortbar“ gekennzeichnet wurden, eventuell bis auf die Frage nach der Zeit.
- Entsprechend der hohen Gewichtung für Zeitangaben in den Stichwortlisten ist die Frage nach dem Zeitpunkt des Termins fast immer beantwortbar, in 94 Prozent der Texte bei den Emails, 90 Prozent bei den leicht gekürzten SMS, 88 bei den mittleren und 91 bei den stark gekürzten Nachrichten. Vom Originaltext abweichende Zeitangaben sind selten, sie entstehen nur, wenn mehrere Zeitangaben vorkommen.
- Auch die Frage nach dem Status des Treffens ist häufig beantwortbar und dies überlebt oft die Kürzung: die Prozentzahlen sind 92 (für die Emails), 86, 79 und 74 (für die verschiedenen SMS-Varianten). Ein Grund ist wohl, dass Verben nicht gestrichen werden, die oft diese Information tragen (`absagen`, `ausfallen`, `vorschlagen` etc.). Die bei dieser Frage ausgewählte Antwort weicht in etwa 35 Prozent der SMS vom Original ab; dies ist jedoch oft der Vielfältigkeit der möglichen Antworten geschuldet, deren Unterschiede (etwa zwischen „ausfallen“ und „absagen“) nicht immer groß sind. Fasst man die möglichen Antworten a) und b) sowie c) und g) zusammen, sinkt die Abweichung auf um 25 Prozent. Bei den anderen Fragen sind abweichende Antworten viel seltener. Insgesamt dürfte die Zahl der wirklichen Missverständnisse gering sein.
- Die Beantwortbarkeit der Fragen 1 bis 5 sinkt im Schnitt von 70 Prozent für die Emails auf 43 Prozent für die stark gekürzten SMS. Dementsprechend steigt der Anteil der Fragen, die für die Email noch beantwortet werden konnten, aber nicht mehr für die SMS, von 16 Prozent bei den leicht gekürzten auf 29 Prozent bei den stark gekürzten SMS.
- Die Informationen zu Art des Treffens und zum Ort sind oft Opfer der Kürzungen. Die Beantwortbarkeit sinkt um 25 bzw. 40 Prozent der Fälle von den Emails zu den stark gekürzten Texten. Zu den leicht gekürzten Texten sinkt die Beantwortbarkeit nur um 12 bzw. 14 Prozent. Die Wörter zur Bezeichnung dieser Informationen

erhalten selten ein höheres Gewicht, lediglich einige häufige Veranstaltungsarten wie `Party` oder `Konferenz` finden sich recht weit oben in den Listen.

- Informationen zu den Teilnehmern eines Treffens sind ohnehin selten, werden dann aber fast immer gestrichen, da zum Beispiel Eigennamen fast nie auf den Stichwortlisten landen.

- Der Durchschnittswert der Antworten zur letzten Frage nach der Verständlichkeit beträgt bei Emails 4.3, bei leichter Kürzung in den SMS 3.1, bei mittlerer Kürzung 2.4 und bei starker 1.8.

Aus diesen Ergebnissen und dem Gesamtbild der Antworten entnehme ich das Fazit, dass die wesentlichen terminbezogenen Informationen, nämlich Zeitpunkt und „Status" des Termins, gut aus leicht gekürzten Sätzen zu entnehmen sind, und dass dies auch für die Art des Termins oft noch gilt; aus stark gekürzten Texten ist jedoch zu häufig kaum noch etwas zu entnehmen bis auf den Zeitpunkt. Die Verständlichkeit der Texte würde durch Kenntnis des Kontextes, die bei den wirklichen Absendern und Empfängern natürlich vorausgesetzt werden kann, oft stark gesteigert, so dass leicht gekürzte SMS-Nachrichten in den meisten Fällen die wesentlichen Informationen werden anbringen können. Bei starker Kürzung wird dies zu oft nicht möglich sein (vgl. die Textbeispiele am Ende des vorigen Kapitels). Die Balance zwischen Unterbringung von viel Information und Lesbarkeit der Texte kann bei 160 Zeichen Längenbeschränkung also nicht immer gehalten werden, sondern muss zugunsten der Lesbarkeit aufgegeben werden.

Insgesamt ist aber demonstriert worden, dass die Bewertung gezielter Zusammenfassungen mit ebenso gezielten Fragen gut möglich ist und ein fundiertes Bild des Informationsgehaltes von Zusammenfassungen liefern kann. Dies bestätigt die Ergebnisse des SUMMAC-Projektes, wo ein ähnliches Vorgehen verwendet wurde. Zu den verwendeten Fragen war dort in jedem Text markiert, welche Passagen die Antwort enthielten. Das beste System erstellte Zusammenfassungen, die im Schnitt 73 Prozent der so markierten Passagen enthielten; allerdings war die Länge der Zusammenfassungen nicht beschränkt, im Gegensatz zu den hier beurteilten SMS-Nachrichten.

6.4. Satzkürzung ohne Filtern

In Abschnitt 5.3 ist ein zusätzliches Experiment beschrieben, bei dem die erste Stufe der Textreduktion übersprungen wird, indem die Satzkürzung direkt zur Anwendung auf alle Sätze kommt. Dazu wird ein Mindestgewicht für Phrasen beliebigen Typs ermittelt, unterhalb dessen eine Phrase entfernt wird (Phrasenfiltern). Die Bewertung der entstehenden Kurztexte konnte aus Zeitmangel nicht in der selben Gründlichkeit erfolgen wie im vorigen Abschnitt, so dass hier meine subjektiven Aussagen über die Lesbarkeit und Verständlichkeit der Texte genügen müssen.

Zunächst können jedoch objektiv wie zuvor die Bewertungsmaße aus Abschnitt 6.2.2 herangezogen werden, die auch hier satzweise bestimmt werden. Ein Satz gilt also als extrahiert, wenn mindestens eine seiner Phrasen über dem Schwellwert für Phrasen liegt. Für die Experimente wurde die Emailsammlung von Terminabsprachen verwendet sowie die Phrasenerkennung mit WAP. Für die Worthäufigkeitsliste sind die zehnfach kreuzvalidierten Resultate 88.8 Prozent Recall und 66 Prozent Precision, also ein F_1-Wert von 75.6 Prozent. Die Accuracy beträgt 92.8 Prozent und der Fallout liegt bei 6.6 Prozent. Die Standardabweichung für alle diese Werte liegt im Bereich der vorigen Experimente. Durchschnittlich bleiben in den Testmengen 1.99 Phrasen pro markiertem Satz erhalten, aber nur 0.08 Phrasen pro nicht markiertem Satz. Weniger als jeder zehnte nicht themenbezogene Satz steuert also eine seiner Phrasen zum reduzierten Text bei. Andererseits bleiben pro themenbezogenem Satz nur zwei Phrasen übrig, während die Durchschnittslänge beliebiger ganzer Sätze (vgl. Seite 44) 4.7 Phrasen beträgt.

In Bezug auf Recall, Precision und Fallout sind diese Ergebnisse sehr zufriedenstellend. Der Precisionwert kann erhöht werden auf über 71 Prozent, indem die Stichwortliste auf 2 Prozent ihrer Länge reduziert wird; dies führt zu noch stärkeren Kürzungen, so dass pro themenbezogenem Satz nur 1.76 Phrasen im Durchschnitt erhalten bleiben. Nach den Ergebnissen zur Lesbarkeit der SMS-Texte ist eine so starke Kürzung jedoch nicht zu empfehlen.

Auch mit der ganzen Liste stellt sich das Problem der Verständlichkeit. Das folgende Beispiel ist zunächst eine gelungene Verkürzung (noch keine SMS) einer Email mit sieben Sätzen, in der Ort und Inhalt eines Treffens diskutiert werden, dessen Zeitpunkt jedoch schon feststeht und in der Verkürzung deutlich bleibt:

(9) `^, das erste Treffen des Themenzirkels findet am 22.12. um 10 Uhr statt.^schlage ich^,.#^, können.`

Wie in Kapitel 5 erläutert, dienen die Zeichen `^` und `#` als Auslassungszeichen. Hier sind aus nicht relevanten Sätzen nur zwei kurze Phrasen erhalten, die natürlich unverständlich sind.

Aber auch themenbezogene Sätze werden oft zu stark gekürzt, wie die zwei folgenden Beispiele deutlich machen.

(10) `^18:00 Uhr.^am Wochenende^(6.12.)^.#^nachmittag.^wir koennen uns erst am Montag^.`

(11) `^, dann Treffen:^Zeit-Mittwoch hängt davon^,.^am Samstag.`

Nach meinem Eindruck ist diese direkte Kürzung nur geeignet, wenn der Leser den Kontext der Texte recht gut kennt und nur spezifische Informationen, im Falle der Terminabsprachen die Zeitangaben, benötigt. Um sanfter zu kürzen, kann man wieder den Schwellwert, hier für ein Phrasengewicht, senken; dabei nimmt die Zahl der unnötigen Phrasen zu, was mit Ausnahme des Email-zu-SMS-Dienstes mit seiner strengen Platzbeschränkung in den meisten Fällen nicht gravierend sein dürfte.

7. Zusammenfassung und Ausblick

Zusammenfassung

Diese Diplomarbeit behandelt gezielte Zusammenfassung von Texten, genauer die Suche nach bestimmten Informationen in beliebigen Texten und die Extraktion der betreffenden Sätze. Zur Vorgabe der interessierenden Informationen ist eine Beispielmenge von Texten notwendig, in der die themenbezogenen Texte markiert sind. Es kann nach den vorgestellten Ergebnissen genügen, nur jeweils jeden Text der Beispielsammlung als zum Thema gehörig oder nicht zu kennzeichnen. Jedoch führt eine Kennzeichnung einzelner Sätze zu besseren Resultaten und ist im Allgemeinen zu bevorzugen.

Aus dem Vergleich der markierten Beispieltexte oder Textteile mit den anderen werden mit verschiedenen Methoden Listen von Stichwörtern errechnet, die das Thema repräsentieren. Die Listen enthalten für jedes Wort ein Gewicht, das annähert, mit welcher Wahrscheinlichkeit (bezogen auf die Beispielmenge) ein Satz, der das Wort enthält, zum Thema gehört. Für beliebige Sätze lässt sich damit ein Gesamtgewicht ermitteln; dies kann auf verschiedene Weisen geschehen, jedoch ist den Experimenten zufolge die Summation der Gewichte aller Wörter des Satzes, geteilt durch die Anzahl der Wörter, die einfachste und erfolgreichste Methode. Das Satzgewicht wird mit einem Schwellwert verglichen; liegt es darüber, so wird der Satz extrahiert. Zur Auswahl eines geeigneten Schwellwertes wird dieser schrittweise von 0 an erhöht. Durch Messung von Recall und Precision, die jeder Schwellwert auf der Beispielmenge erbringt, wird der beste Schwellwert ermittelt. Auf diese Weise wird der Schwellwert an die gegebene Liste angepasst.

Die Bewertung dieses Extraktionsverfahrens erfolgte im wesentlichen mit Recall- und Precisionangaben auf zur Beispielmenge disjunkten Testmengen. Zwei verschiedene Textsammlungen wurden verwendet, das Szenario ist das des Information Filtering. Für Terminabsprachen mit Emails liegt das beste Ergebnis bei knapp über 80 Prozent nach dem F_1-Maß (ein bestimmter Mittelwert aus Recall und Precision). Dieses sehr gute Resultat konnte auf der zweiten Textsammlung nicht erreicht werden, aber ein Wert von über 70 Prozent F_1 zeigt, dass das Extraktionsverfahren auch hier funktioniert. Als wichtige Option besteht die Möglichkeit, über eine Senkung des Schwellwertes das Verhältnis von Recall und Precision zugunsten des Recalls zu ändern. Die Experimente haben gezeigt, dass dabei der Precisionwert hoch genug bleibt, um eine deutliche Erleichterung bei der Suche nach Informationen bieten zu können. Insbesondere wurde bei jedem Experiment (für beide Textsammlungen) mit Hilfe des Falloutwertes verdeutlicht, dass die große Mehrheit der nicht relevanten Texte nicht gelesen werden muss, wenn diese Art der Filterung angewandt wird, und dies gilt auch bei erhöhtem Recall.

Zusätzlich zur Satzextraktion wurde eine zweite Stufe der Kürzung entwickelt, die einzelne Sätze kürzt. Als Hinweis auf Satzteile, die in Bezug auf das vorgegebene Thema wichtig sind, dienen die gleichen Wortlisten wie zur Extraktion, beziehungsweise ihr oberer Teil. Nicht themenbezogene Satzteile werden gestrichen. Die Kürzung erfolgt abgestuft nach Radikalität. Niedrige Stufen ersetzen einzelne Wörter durch ihre Abkürzungen und entfernen Wörter, die zumeist inhaltsleer sind. Höhere Stufen beruhen auf der Erkennung grammatisch zusammengehöriger Phrasen durch einen Parser für die deutsche Sprache. Dabei werden Phrasen entfernt, wenn sie kein Wort aus der Wortliste enthalten. Die Satzkürzung kann mit gewissen Veränderungen auch direkt auf ungefilterten Text angewendet werden (Phrasenfiltern).

Im Rahmen eines simulierten Email-zu-SMS-Dienstes wurde der Informationsgehalt der erzeugten SMS-Nachrichten mit Hilfe von Testpersonen bewertet. Die SMS-Texte wurden mit drei verschiedenen Satzkürzungsstufen erstellt und die Beantwortbarkeit von gezielten Fragen bei Kenntnis der verschiedenen Texte beurteilt. Die gezielte Fragestellung war möglich wegen der Vorgabe eines Themas, so dass der Informationsgehalt der Kurztexte gut untersucht werden konnte. Gleichzeitig konnte der Tradeoff zwischen Unterbringung von möglichst viel Information und Lesbarkeit der entstehenden, gekürzten SMS-Texte beurteilt werden. Den Ergebnissen zufolge enthalten die meisten SMS-Nachrichten die wesentlichen themenbezogenen Informationen, es können jedoch auch wichtige Informationen verloren gehen. Die Texte, die auf der stärksten Satzkürzung basieren, sind zu oft unverständlich, während leicht gekürzte Texte meistens gut lesbar sind; auch wenn sie weniger Informationen enthalten mögen, sind sie daher vorzuziehen.

Beurteilung und Ausblick

Zunächst kann festgehalten werden, dass es in dieser Arbeit gelungen ist, die in der Einleitung aufgestellten Fragen zu beantworten. Die Ermittlung von für ein bestimmtes Thema charakteristischen Wortlisten ist möglich, wenn entsprechend gekennzeichnete Beispieltexte vorliegen. Mit diesen Wortlisten können die themenbezogenen Segmente längerer Texte gut aufgefunden werden, bis zu 80 Prozent der entsprechenden Stellen, satzweise gemessen, wurden erkannt. Die Kürzung von Sätzen wurde erhellt, ebenso konnte die Bewertung des Informationsgehaltes der entstehenden Kurztexte dank ihrer Themenfokussierung fundiert durchgeführt werden.

Die vorgestellte Methode der Satzextraktion unterscheidet sich von den meisten bisherigen Verfahren durch ihre Ausrichtung auf ein bestimmtes, implizit in den Markierungen vorgegebenes Thema. Auf dem Gebiet der automatischen Textzusammenfassung stand zunächst die Extraktion von Sätzen, die für ihren gesamten Text repräsentativ sind, im Vordergrund (generische Extraktion). Forschungen an themenbezogenen Zusammenfassungen kamen hinzu, erweiterten in den meisten Fällen jedoch nur die bisherigen Extraktionsmethoden um einen Aspekt, der die Themenvorgabe mit einfließen lässt, etwa die Übereinstimmung von Wörtern der Sätze mit Wörtern aus einer Benutzeranfrage wie im Information Retrieval, die das Thema vorgibt. Dagegen verwendet das hier vorgestellte Verfahren *ausschließlich* die wortweise Übereinstimmung mit den Listen, die dafür das vorgegebene Thema deutlich genauer repräsentieren als eine kurze Benutzeranfrage. Ähnlich stark wortbezogen sind noch die Lexical-Cohesion-Verfahren, die in Abschnitt 3.5.1 erläutert werden. Diese verwenden jedoch explizit modellierte semantische Beziehungen zwischen Wörtern, was hier keine Rolle spielt. Statt dessen wird eine semantische Beziehung zwischen den hochgewichteten Wörtern implizit unterstellt, weil sie für das gleiche Thema charakteristisch sind und derselben zugrunde liegenden Textsammlung entstammen.

Gezielte Extrakte sind im Allgemeinen einfacher zu erstellen als generische, da genauere Kriterien für die Auswahl von Textteilen vorliegen. Diese Arbeit hat gezeigt, dass für diese Auswahl Wortlisten ausreichen können, deren Wörter das gewünschte Thema repräsentieren. Wie diese Listen ermittelt werden, ist eine davon unabhängige Frage. Sie könnten auch von Hand erstellt werden, allerdings können dabei leicht wichtige Wörter übersehen oder Wörter in ihrer Bedeutung überschätzt werden. Die automatische Ermittlung bietet hier große Vorteile, zumal sich die Listen damit an die Charakteristik verschiedener Textsammlungen anpassen lassen. Die Notwendigkeit der Markierung von themenbezogenen Texten oder Textteilen stellt dabei einen gewissen Aufwand dar. Es wurde gezeigt, dass die textweise Markierung ausreichen kann, aber nicht immer ausreicht; unter Berücksichtigung der statistischen Natur der vorgestellten Methoden zur Listenberechnung kann erwartet werden, dass auch die textweise berechneten Listen besser werden, je größer die verwendete Textsammlung ist. Wegen des nicht ausreichenden Resultats mit textweise berechneten Listen beim Wahlergebniskorpus muss jedoch davon ausgegangen werden, dass mindestens einige hundert Texte für gute Ergebnisse notwendig sind. Es liegt nahe, dies mit Hilfe bekannter, bereits textweise markierter Korpora zu überprüfen, wie sie bei vielen Untersuchungen zur Textklassifikation eingesetzt worden sind (siehe zum Beispiel [Yang und Liu 1999] oder [Han und Karypis 2000]). Bei satzweiser Markierung sinkt die Mindestgröße des Korpus, während der Aufwand deutlich steigt. Wünschenswert wäre aber, dass ein Benutzer das System schnell und einfach an seine Bedürfnisse, also für ihn wichtige Themen, anpassen kann. Im Information Filtering-Szenario sollte dazu ein Benutzer zum Training des Systems die täglich anfallenden Texte nach Interesse markieren, und mit fortschreitender Zeit sollte das System die Filterung mehr und mehr verbessern können, indem die Wortlisten immer genauer werden. Mit einer geeigneten Umgebung, die das Markieren eines Textteils auf einen Mausklick reduziert, könnte der Trainingsaufwand zumutbar sein, was aber von der Art der Texte und des interessierenden Themas abhängt, wie diese Arbeit gezeigt hat. Einen Test wert wäre die Markierung von Absätzen statt Sätzen zur Verringerung des Markierungsaufwandes.

Vielleicht kann aber die Ermittlung der Wortlisten von einer Beispieltextmenge unabhängig gemacht werden. Oben wurde die Verwendung explizit modellierter semantischer Wortbeziehungen angesprochen, ein Beispiel sind Lexical Chains. Diese wurden bisher nur zur generischen Extraktion eingesetzt, ihre Verwendung zur gezielten Themensuche ist jedoch ein nicht von der Hand zu weisender Ansatz. Auf der Grundlage eines elektronischen Thesaurus könnten mit wenigen Ausgangswörtern weitere themenbezogene Wörter gefunden werden. Andererseits ist fraglich, ob auf diese Weise Wörter wie `Termin` oder `treffen` mit Zeitangaben wie `Montag` oder `nachmittag` in Verbindung gebracht würden, wie es etwa durch die Worthäufigkeitsliste bei der Emailsammlung geschah. Die in WordNet bzw. GermaNet verwendeten semantischen Beziehungen (Abschnitt 3.5.1) leisten dies nicht. Möglich wäre dies aber durch eine statistische Auswertung von gemeinsam (im gleichen Text) auftretenden Wörtern in großen Korpora (Lexical Cohesion), wie in der auf Seite 28 zitierten Arbeit [Baldwin und Morton 1998]. Hier wird eine unmarkierte Beispielmenge verwendet; diese ist in Gestalt öffentlich zugänglicher Korpora—abhängig von der gewünschten Sprache—zumeist verfügbar. In beiden

Fällen (thesaurus- und korpusbasierte Wortlisten) hat man jedoch nicht unbedingt die Anpassung an die Textsammlung, auf die das Verfahren angewendet werden soll, deren vielleicht eigene Charakteristik ausgenutzt werden könnte. Insgesamt wäre die Durchführung entsprechender Experimente zu diesen Vorschlägen sehr interessant, gute Resultate sind durchaus möglich, insbesondere da bei einigen der hier durchgeführten Experimente schon wenige Stichwörter für zufriedenstellende Extraktionsergebnisse gesorgt haben.

Naheliegende weitere Tests ergeben sich dadurch, dass die vorgestellten Methoden in dieser Arbeit nur für die deutsche Sprache getestet wurden und nur auf zwei Textsammlungen. Weitere Sprachen, Textgenres und inhaltliche Gebiete müssen hinzu kommen, um die breite Anwendbarkeit der Methoden zu überprüfen. Interessant wäre dabei die Frage, ob die Stammformenreduktion bei allen Sprachen einen ähnlichen Vorteil erzielt wie im Deutschen; flektionsarme Sprachen wie Englisch mögen auch ohne sie auskommen. Über die Extraktion mit Wortlisten hinaus könnte man Benutzern des Systems zusätzliche Optionen bieten, wie die Filterung von Sätzen, die in einer bestimmten Zeitform stehen (da etwa vergangene Termine nicht mehr interessant sind) oder die unbedingte Extraktion von Textsegmenten, in denen bestimmte Wörter auftreten, zum Beispiel Eigennamen. Solche einfachen Einstellungen können vom Benutzer für jede Anwendung selbst vorgenommen werden.

Weiter ist zu beachten, dass die Methoden zur Satzauswahl dieser Diplomarbeit auch mit mehreren Wortlisten, die zu verschiedenen Themen erstellt wurden, funktionieren würden. Dazu würde jeder Satz, dessen Gesamtgewicht bezogen auf eine der Listen über ihrem zugehörigen Schwellwert liegt, extrahiert, womit eine gleichzeitige Suche nach mehreren Themen möglich ist. Alternativ kann eine Liste für mehrere Themen erstellt werden, indem die Markierung der Trainingsmenge alle interessierenden Themen einschließt. Der Austausch einzelner Themen zwischen mehreren Benutzern des Systems ist damit ebenso leicht möglich wie die Einstellung eines Systems von vorneherein auf mehrere Interessen seines Benutzers. Damit gewinnt das System den Charakter eines allgemeinen Informationsfilters, wie oben schon angedeutet wurde. Gesammelt werden dann nicht mehr Wörter zu einem bestimmten Thema, sondern Wörter, die die Interessenlage des Benutzers allgemein wiedergeben. Durch ständiges Feedback seitens des Benutzers, in Form der Markierung der Texte (oder eventuell Textteile) als interessant oder nicht, kann die Filterung auch an zeitlich wechselnde Interessenlagen angepasst werden (dazu muss den Benutzern allerdings auch die Möglichkeit gegeben sein, neue Interessen durch aktive Auswahl bisher weggefilterter Texte anzuzeigen). Dieses Szenario erinnert an Informationsfilter wie die Persönliche Zeitung in [Veltmann 1997], bei denen ebenfalls ein Benutzerfeedback für die Auswahl von Texten sorgt, die zur Zeit den Interessen des Benutzers entsprechen. Dank der dort erzielten Erfolge scheinen Experimente in dieser Richtung gerechtfertigt und vielversprechend, zumal die Erstellung von Wortlisten nach den vorgestellten Verfahren sehr effizient möglich ist, also den Aufwand für Filtersysteme senken könnte. Die Extraktion von Textsegmenten ist dann nicht mehr an einzelnen Themen ausgerichtet, sondern an größeren Interessensgebieten. Wenn gewünscht, können auch ganze Texte mit den Wortlisten klassifiziert werden, wenn eine Extraktion zum Zwecke der Verkürzung in der entsprechenden Anwendung überflüssig ist; die Klassifikation ganzer Emails mit Wortlisten übertraf in dieser Arbeit die herkömmlichen Textklassifikationsverfahren (Abschnitt 6.2.3). Auch dies sollte mit den erwähnten Korpora aus der Textklassifikation überprüft werden. Sollten die Ergebnisse überzeugen, so kann das Wortlistenverfahren überall dort eingesetzt werden, wo Textklassifikation sinnvoll ist, zum Beispiel bei der Sortierung von Webseiten in Webverzeichnissen.

Ein zusätzliches Feld für weitergehende Untersuchungen bietet die Satzkürzung aus Kapitel 5. Wie dort erläutert ist, existieren sehr wenige Arbeiten auf diesem Gebiet, insbesondere die Bewertung verschiedener Kürzungsverfahren gestaltet sich schwierig. Um so eine Bewertung vergleichsweise objektiv durchführen zu können, wäre ein annotierter Korpus von Sätzen notwendig, in denen die überflüssigen Teile markiert sind. Dann könnten verschiedene Kürzungsverfahren auf diesen Daten getestet werden. Da es aber von den Lesern und ihrem momentanen Interesse abhängt, welche Teile wichtiger sind als andere, müssen die Kriterien für einen überflüssigen Satzteil in so einer Testsammlung genau festgelegt sein, damit die Kürzungsverfahren sich darauf einstellen können. Die Arbeit mit einem solchen Korpus setzt gute und robuste sprachverarbeitende Werkzeuge voraus. Auch die Kürzungsverfahren aus den Abschnitten 5.1 und 5.3 würden von einem verbesserten Parser profitieren, ebenso wie von einem Tagger und einem Satzparser.

Zur Satzkürzung erscheinen rein syntaxbasierte Ansätze zunächst wenig vielversprechend, da die wichtige Information in sehr verschiedenen syntaktischen Konstruktionen vorkommen kann (Abschnitt 5.1). Immerhin könnte es sein, dass eine bestimmte Art von Information *meistens* in wenigen bestimmten syntaktischen Konstruktionen steckt; dies wäre abhängig von der Art der Texte wie auch der gesuchten Information. Sicherlich sind bestimmte Strukturen wie Subjekt und Verb wichtiger als andere. Sollte es solche nicht auf den ersten Blick sichtbare Zusammenhänge geben, so wären maschinelle Lernverfahren in Verbindung mit einem guten Parser der richtige Weg, sie zu finden. Dabei müsste man sicherlich in Kauf nehmen, dass die Information in einigen Fällen gerade nicht dort steckt, wo man sie aufgrund des syntaktischen Patterns vermutet. Um ein Lernverfahren umzusetzen, wäre wieder die Kennzeichnung der interessierenden Information in jedem Satz einer Beispieltextmenge nötig, was einen recht hohen Aufwand darstellt, der für jedes gewünschte Thema wiederholt werden müsste. Als Lernaufgabe kann erstens die Klassifikation syntaktischer Teilstrukturen in themenbezogen oder nicht in Betracht gezogen werden; zweitens, mit höherem Aufwand, aber aussichtsreicher, könnte versucht werden, eine Zuordnung von kompletten zu gekürzten Syntaxbäumen zu lernen.

Die Ermittlung von Wortlisten in dieser Diplomarbeit bietet aber die Möglichkeit einer Satzkürzung, die sich genauer am Inhalt eines Satzes orientiert, indem die Wörter der Listen als semantische Hinweise zur Kürzung verstanden werden. Das durch die Wortlisten gegebene Gewicht könnte nun bei den obigen Lernansätzen als zusätzliches Attribut verwendet werden, was die Erfolgsaussichten sicherlich steigert. Es wäre dann interessant zu untersuchen, welchen jeweiligen Einfluss die syntaktischen und die semantischen Attribute auf das Lernergebnis haben.

Abschließend zusammengefasst, hat diese Diplomarbeit einige neue Ansätze zur flexiblen, effizienten und gezielten Suche nach Informationen in Texten aufgezeigt, die nur über Stichwortlisten erfolgt, für deren automatische Ermittlung es mehrere, zum Teil noch zu erprobende Möglichkeiten gibt. Die Stichwortlisten bieten ferner Ansätze zur bisher kaum erforschten Kürzung einzelner Sätze um irrelevante Teile.

Anhang A: Wortlisten

A.1. Füllwörter

Die Liste der Füllwörter, die auf Stufe drei der Satzkürzungen entfernt werden (siehe Abschnitt 5.1), ist hier wiedergegeben.

`es`	`eigentlich`	`mal`	`eben`
`überhaupt`	`schon`	`also`	`ganz`
`nämlich`	`naja`	`na`	`nun`
`halt`	`hiermit`	`einmal`	`konkret`
`wieder`	`doch`	`recht`	`natürlich`
`sehr`	`wohl`	`erst`	`noch`
`auch`	`ja`		

A.2. Stichwortliste zu Terminabsprachen

Die folgende Stichwortliste zum Thema „Terminabsprachen“ wurde mit der G^2-Methode bestimmt. Sie enthält nur Stammformen und basiert auf der satzweisen Markierung der Trainingsmenge. Sie enthält ebenso wie die Liste aus Abschnitt 6.2.3 833 Wörter, da sie auf der gleichen Trainingsmenge basiert. Die ersten 20 sind:

1	dayofweek	1.0000	11	raum	0.1398
2	date	0.9837	12	treff	0.1292
3	uhr	0.8638	13	monthofyear	0.0880
4	termin	0.6333	14	statt	0.0868
5	time	0.6095	15	stattfind	0.0740
6	am	0.4326	16	ok	0.0734
7	cluster	0.3346	17	nachmittag	0.0688
8	Number	0.2981	18	einlad	0.0604
9	um	0.1684	19	nah	0.0591
10	treffen	0.1669	20	schlag	0.0586

Besonderheiten zu einigen Wörtern sind im Abschnitt 6.2.3 erläutert. Das Wort `cluster` ist ein weiterer Platzhalter für Folgen von Zahlen, denen MESON keine Zeit-oder Datumsinterpretation zuordnen kann. `Number` steht für einzelne Zahlen. Das Wort `schlag` erhält sein Gewicht durch Ausdrücke wie `Ich schlage als Termin ... vor`, die in der Sammlung recht häufig sind. Interessant ist das hohe Gewicht von `am` und `um`, das aus ihrer Verwendung in Zeitangaben folgt. Es folgt ein weiter unten gelegener Ausschnitt derselben Liste.

200	gemeinsam	0.0014	210	wohnung	0.0008
201	ende	0.0014	211	leicht	0.0008
202	bis	0.0013	212	anschliess	0.0008
203	vergess	0.0012	213	eher	0.0008
204	geh	0.0011	214	anmeld	0.0008
205	bochum	0.0010	215	bereit	0.0008
206	teilnehm	0.0010	216	regel	0.0008
207	esse	0.0009	217	wart	0.0008
208	klein	0.0008	218	lieber	0.0008
209	zimmer	0.0008	219	komm	0.0008

Hier ist das Wort `teilnehmen` noch ein Wort, das mit Treffen von Personen assoziiert werden kann, ebenso wie `kommen`, aber beide werden natürlich auch in anderen Zusammenhängen verwendet. Das Ende einer Stichwortliste enthält dann nur noch allgemeine Wörter:

820	werd	0.0000	827	da	0.0000
821	ueber	0.0000	828	hab	0.0000
822	woll	0.0000	829	angebot	0.0000
823	ander	0.0000	830	bei	0.0000
824	unter	0.0000	831	ps	0.0000
825	mit	0.0000	832	bernd	0.0000
826	sei	0.0000	833	gut	0.0000

Dass alle diese Wörter das Gewicht 0 haben, liegt daran, dass sie vor der Skalierung auf den Bereich 0 bis 1 das gleiche Gewicht hatten wie das letzte Wort, welches durch die Skalierung immer das Gewicht 0 erhält. (Die Skalierung erfolgt, indem von jedem Wortgewicht das kleinste Gewicht der Liste abgezogen wird und das Ergebnis durch die Differenz von größtem und kleinstem Gewicht der Liste geteilt wird.)

A.3. Stichwortlisten zu Wahlergebnissen

Die folgende Stichwortliste zum Thema „Wahlergebnisse“ wurde mit der Worthäufigkeitsmethode auf Basis der satzweisen Markierung berechnet; es sind nur Stammformen enthalten. Insgesamt enthält die Liste 995 Wörter. Es ist nur der obere Teil abgedruckt.

1	gewinn	1.0000	16	endergebnis	0.2381
2	huerde	0.4643	17	sitz	0.2292
3	erring	0.4524	18	stimme	0.2257
4	mandat	0.3626	19	verzeichn	0.2143
5	triumph	0.3571	20	hervorgeh	0.2143
6	buess	0.3571	21	sack	0.2143
7	freiheitlich	0.3571	22	einzieh	0.2143
8	verfehl	0.3214	23	rechnung	0.1939
9	absolut	0.2959	24	vorn	0.1786
10	dell	0.2857	25	zuleg	0.1786
11	freiheit	0.2857	26	erziel	0.1758
12	stimmberechtigt	0.2857	27	liberal	0.1714
13	beziehungsweise	0.2857	28	sieger	0.1667
14	vorlaeufig	0.2678	29	beteiligung	0.1619
15	verfueg	0.2619	30	prognostizier	0.1428

Das Wort `dell` ist Bestandteil eines Namens einer italienischen Partei, von deren Wahlsieg zwei Texte der Sammlung berichten. Das Wort `buess` ist Stammform von `einbüssen`, `sack` ist Stammform von `absacken`; beide Vokabeln treten häufig bei Vergleichen von aktuellen Wahlergebnissen mit früheren Wahlen auf. Das Wort `beziehungsweise` wird oft benutzt, um mehrere Prozentangaben oder Prozent- mit Sitzanteilangaben zu verbinden.

Die folgende Liste entstand aus textweiser Markierung. Wie auf Seite 79 geschildert, enthält sie mehr allgemeine politische Begriffe.

1	demokratisch	1.0000	16	sitz	0.4079
2	parlament	0.8285	17	mehrheit	0.4069
3	stimme	0.8285	18	wahl	0.3908
4	demokrat	0.6856	19	konservativ	0.3711
5	bildung	0.6570	20	fuehrung	0.3711
6	wahlsieg	0.6284	21	beobachter	0.3711
7	fdp	0.5855	22	anhaenger	0.3711
8	erziel	0.5569	23	gewinn	0.3711
9	unabhaengigkeit	0.5427	24	beteiligung	0.3521
10	waehler	0.5284	25	labour	0.3426
11	zdf	0.5141	26	nachrichtendienst	0.3426
12	premier	0.5141	27	niederlage	0.3426
13	verlierer	0.5141	28	ard	0.3140
14	link	0.5141	29	enttaeuschung	0.3140
15	abschneid	0.4283	30	opposition	0.3140

Anhang B: Hinweise zur Implementation

In diesem Anhang werden ein paar kurze Hinweise zur Implementation gegeben, die vor allem für interessierte Anwender des Programms oder einzelner seiner Module am Lehrstuhl für Künstliche Intelligenz der Universität Dortmund gedacht sind. Ich weise darauf hin, dass das Programm der Universität Dortmund für Zwecke der Forschung und Lehre zur Verfügung steht, ich mir jedoch alle weiteren Rechte vorbehalte.

Das Programm wurde in der Programmiersprache Java implementiert und folgt der objektorientierten Denkweise. Alle Klassen und Methoden sind vollständig auf Deutsch dokumentiert nach den Javadoc-Konventionen. Die einzelnen Module des Programms sind in Klassen zusammengefasst, deren Methoden die verschiedenen Funktionen des Moduls zur Verfügung stellen. Alle Klassen gehören zu einem Package namens `da`. Die zentrale Datenstruktur wird von der Klasse `Email` zur Verfügung gestellt, deren Objekte alle Sätze eines Textes mitsamt ihrer Analyse als Phrasen, die aus Wörtern aufgebaut sind, sowie deren Markierungen im Korpus und durch das verwendete Extraktionsverfahren halten. Dazu kommen im Falle von Emails Metadaten wie Absender und Betreffzeile.

Die Erstellung der Ranglisten erfolgt mit den Methoden der Klasse `WortRanker`. Die Auswahl von Sätzen über den Schwellwert kann mit der Klasse `RelevanteStellenFinder` durchgeführt werden. Alle Verfahren zur Klassifikation von Sätzen sind in der Klasse `Klassifikator` zusammengefasst. Die Satzkürzung erfolgt mit Methoden der Klasse `Kuerzer`. Die Auswertung von Testmengen geschieht mit `Bewertung`; dabei wird zur Verringerung des Speicheraufwandes auf gesonderte Objekte zurückgegriffen, die für jeden Text nur die vorgegebene und die vom Verfahren ermittelte Klassifikation enthalten. Die Klassen `Verarbeitung` und `Steuerklasse` stellen übergeordnete Methoden zur Steuerung der Verfahren im Zusammenhang zur Verfügung.

In der Klasse `Vorfilter` findet sich die Vorverarbeitung von Emails, die die Metadaten extrahiert. Außerdem wird hier die Markierung der Texte berücksichtigt, die erfolgt, indem zu markierende Passagen im Originaltext mit dem Tag `$T$` eingerahmt werden.

Trotz der Plattformunabhängigkeit der Programmiersprache läuft die derzeitige Version nur unter Unix bzw. Solaris, weil verschiedene Werkzeuge verwendet werden, die zum Teil nur auf dieser Plattform laufen. Die verwendeten Werkzeuge sind: Meson und Wap (Kapitel 2), Tcat (Abschnitt 4.2.4) und mySVM (Abschnitt 6.2.3). Tcat dient der Auszählung von Worthäufigkeiten in verschiedenen Teilen des Korpus oder einzelnen Texten. Die Einbindung der Werkzeuge in den Programmcode ist unvermeidbar an einigen Stellen etwas unübersichtlich.

Für die Steuerung des Programms muss eine Konfigurationsdatei angelegt werden, die mitgelieferte Beispieldatei namens `konfig.kfg` ist ausführlich kommentiert. Darin werden Arbeitsverzeichnisse und verschiedene Parameter festgelegt.

Literaturverzeichnis

[Abney 1996] Abney, S. (1996). *Partial Parsing via Finite-State Cascades.* In: Carroll, John, Hrsg.: *Proceedings of ESSLLI'96 Robust Parsing Workshop*, S. 8–15, Prag, Tschechische Republik.

[Abney 1997] Abney, Steven (1997). *Part of Speech-Tagging and Partial Parsing.* In: Bloothooft, G. und S. Young, Hrsg.: *Corpus-Based Methods in Language and Speech Processing*, Kap. 4, S. 118–136. Kluwer Academic Publishers, Dordrecht.

[Abu-Hakima et al. 1996] Abu-Hakima, Suhayya, R. Liscano und R. Impey (1996). *Cooperative Agents That Adapt for Seamless Messaging in Heterogeneous Communication Networks.* In: Imam, Ibrahim, Hrsg.: *Working Notes of the AAAI-96 Workshop on Intelligent Adaptive Agents*, Portland, OR.

[Appelt und Israel 1999] Appelt, D. und D. Israel (1999). *Introduction to Information Extraction Technology.* Tutorial for IJCAI-99, Stockholm.

[Arbanowski und van der Meer 1999] Arbanowski, S. und S. van der Meer (1999). *Service Personalization for Unified Messaging Systems.* In: *Proceedings of The Fourth IEEE Symposium on Computers and Communications.*

[Baeza-Yates und Ribeiro-Neto 1999] Baeza-Yates, Ricardo und B. Ribeiro-Neto (1999). *Modern Information Retrieval.* Addison-Wesley-Longman Publishing Co.

[Baldwin und Morton 1998] Baldwin, Breck und T. S. Morton (1998). *Dynamic Coreference-Based Summarization.* In: *Proceedings of the Third Conference on Empirical Methods in Natural Language Processing (EMNLP-3)*, Granada, Spanien.

[Barzilay und Elhadad 1999] Barzilay, Regina und M. Elhadad (1999). *Using Lexical Chains for Text Summarization.* In: Mani, Inderjeet und M. T. Maybury, Hrsg.: *Advances in Automatic Text Summarization*, S. 111–121. MIT Press, Cambridge, MA.

[Brandow et al. 1995] Brandow, R., K. Mitze und L. Rau (1995). *Automatic Condensation of Electronic Publications by Sentence Selection.* Information Processing and Management, 31(5):675–685.

[Brill 1995] Brill, E. (1995). *Transformation-Based Errordriven Learning and Natural Language Processing: A Case Study in Part of Speech-Tagging.* Computational Linguistics, 21(4):543–566.

[Burges 1998] Burges, C. (1998). *A Tutorial on Support Vector Machines for Pattern Recognition*. Data Mining and Knowledge Discovery, 2(2):121–167.

[Burke et al. 1997] Burke, Robin, K. Hammond, V. Kulyukin, S. Lytinen, N. Tomuro und S. Schoenberg (1997). *Question Answering from Frequently-Asked Question Files: Experiences with the FAQ Finder System*. Technischer Bericht, University of Chicago, Department of Computer Science.

[Busemann et al. 1997] Busemann, Stephan, T. Declerck, A. K. Diagne, L. Dini, J. Klein und S. Schmeier (1997). *Natural Language Dialogue Service for Appointment Scheduling Agents*. Technischer Bericht RR-97-02, Deutsches Forschungszentrum für Künstliche Intelligenz, Saarbrücken.

[Buyukkokten et al. 2000] Buyukkokten, Orkut, H. Garcia-Molina und A. Paepcke (2000). *Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices*. In: *Proceedings of the Tenth International World Wide Web Conference*.

[Chen 1995] Chen, Hsinchun (1995). *Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms*. Journal of the American Society of Information Science, 46(3):194–216.

[Cohen 1995] Cohen, J.D. (1995). *Highlights: Language- and Domain-Independent Automatic Indexing Terms for Abstracting*. Journal of the American Society for Information Science, 46(3). Man beachte auch das Erratum in Bd. 47, 3, S.260.

[Cortes und Vapnik 1995] Cortes, Corinna und V. N. Vapnik (1995). *Support–Vector Networks*. Machine Learning Journal, 20:273–297.

[Cowie und Lehnert 1996] Cowie, Jim und W. Lehnert (1996). *Information Extraction*. Communications of the ACM, 39(1):80–91.

[Culy 1985] Culy, Christopher (1985). *The Complexity of the Vocabulary of Bambara*. Linguistics and Philosophy, 8(3):345–351.

[Damashek 1995] Damashek, M. (1995). *Gauging Similarity via n-Grams: Text Sorting, Categorization and Retrieval in Any Language.*. Science, 267:843–848.

[Earley 1987] Earley, Jay (1987). *An Efficient Context–Free Parsing Algorithm*. In: Grosz, Barbara J., K. Sparck-Jones und B. L. Webber, Hrsg.: *Readings in Natural Language Processing*, Kap. 1, S. 25–33. Morgan Kaufmann, Los Altos, CA.

[Frank et al. 1999] Frank, Eibe, G. W. Paynter, I. H. Witten, C. Gutwin und C. G. Nevill-Manning (1999). *Domain-Specific Keyphrase Extraction*. In: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, S. 668–673, Kalifornien. Morgan Kaufmann.

[Glickman und Jones 1999] Glickman, Oren und R. Jones (1999). *Examining Machine Learning for Adaptable End-to-End Information Extraction Systems*. In: *Proceedings of the AAAI 1999 Workshop on Machine Learning for Information Extraction*.

[Goldstein et al. 1999] Goldstein, Jade, M. Kantrowitz, V. Mittal und J. Carbonell (1999). *Summarizing Text Documents: Sentence Selection and Evaluation Metrics*. In: *Proceedings of ACM-SIGIR'99*, S. 121–128.

[Grefenstette und Tapanainen 1994] Grefenstette, Gregory und P. Tapanainen (1994). *What is a Word, What is a Sentence? Problems of Tokenisation*. In: *Proceedings of the 3rd Conference on Computational Lexicography and Text Research, COMPLEX'94*, Budapest.

[Hahn und Mani 1998] Hahn, Udo und I. Mani (1998). *Automatic Text Summarization*. Tutorial for the Fifteenth National Conference on Artificial Intelligence (AAAI), Madison, Wisconsin.

[Hamp und Feldweg 1997] Hamp, B. und H. Feldweg (1997). *GermaNet – a Lexical-Semantic Net for German*. In: Vossen, P., N. Calzolari, G. Adriaens, A. Sanfilippo und Y. Wilks, Hrsg.: *Proceedings of the ACL/EACL-97 Workshop on Automatic Information Extraction and Building Lexical Semantic Resources for NLP applications*, Madrid.

[Han und Karypis 2000] Han, Eui-Hong und G. Karypis (2000). *Centroid-Based Document Classification: Analysis and Experimental Results*. Technischer Bericht 00-017, University of Minnesota, Department of Computer Science, Minneapolis, MN 55455.

[Hand und Sundheim 1998] Hand, T. Firmin und B. Sundheim, Hrsg. (1998). *TIPSTER-SUMMAC Summarization Evaluation. Proceedings of the TIPSTER Text Phase III Workshop*, Washington.

[Hearst und Pedersen 1996] Hearst, Martin A. und J. O. Pedersen (1996). *Re-examining the Cluster Hypothesis: Scatter/Gather on Retrieval Results*. In: *Proceedings of the Nineteenth Annual International ACM SIGIR Conference*, Zürich.

[Helbig und Mertens 1994] Helbig, Hermann und A. Mertens (1994). *Word Agent Based Natural Language Processing*. In: Boves, Loe und A. Nijholt, Hrsg.: *Speech and Language Engineering, Proceedings of the 8th Twente Workshop on Language Technology*, S. 65–74, Enschede, NL. Faculteit Informatica, Universiteit Twente.

[Hovy und Marcu 1998] Hovy, Eduard und D. Marcu (1998). *Automated Text Summarization Tutorial*. Tutorial at the COLING/ACL.

[Iwayama und Tokunaga 1995] Iwayama, Makoto und T. Tokunaga (1995). *Cluster-Based Text Categorization: a Comparison of Category Search Strategies*. In: Fox, Edward A., P. Ingwersen und R. Fidel, Hrsg.: *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, S. 273–281, Seattle, US. ACM Press, New York.

[Jacobs und Rau 1993] Jacobs, Paul S. und L. F. Rau (1993). *Innovations in Text Interpretation*. Artificial Intelligence, 63:143–191.

[JING et al. 1998] JING, H., R. BARZILAY, K. MCKEOWN und M. ELHADAD (1998). *Summarization Evaluation Methods: Experiments and Analysis*. In: *Working Notes of the AAAI Spring Symposium on Intelligent Text Summarization*, S. 60–68. AAAI.

[JOACHIMS 1997] JOACHIMS, T. (1997). *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. LS8-Report 23, Universität Dortmund, LS VIII-Report.

[JOACHIMS 1998] JOACHIMS, T. (1998). *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. In: *Proceedings of the European Conference on Machine Learning*, S. 137 – 142, Berlin. Springer.

[JOACHIMS 2001] JOACHIMS, THORSTEN (2001). *The Maximum-Margin Approach to Learning Text Classifiers: Methods, Theory, and Algorithms*. Doktorarbeit, Fachbereich Informatik, Universität Dortmund.

[KENNEDY und BOGURAEV 1996] KENNEDY, CHRISTOPHER und B. BOGURAEV (1996). *Anaphora for Everyone: Pronominal Anaphora Resolution Without a Parser*. In: *Proceedings of the Sixteenth COLING*, S. 113–118, Kopenhagen, Dänemark.

[KLINKENBERG 1998] KLINKENBERG, RALF (1998). *Maschinelle Lernverfahren zum adaptiven Informationsfiltern bei sich verändernden Konzepten*. Diplomarbeit, Universität Dortmund, Fachbereich Informatik.

[KOSKENNIEMI 1984] KOSKENNIEMI, K. (1984). *A General Computational Model for Word-form Recognition and Production*. In: *Proceedings of the 10th International Conference on Computational Linguistics and the 22nd Annual Meeting of the Association for Computational Linguistics (COLING)*, S. 178–181, Stanford University, Kalifornien.

[KUPIEC et al. 1995] KUPIEC, JULIAN, J. PEDERSEN und F. CHEN (1995). *A Trainable Document Summarizer*. In: *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, S. 68–73, Seattle, Washington.

[LAPPIN und LEASS 1994] LAPPIN, SHALOM und H. LEASS (1994). *An Algorithm for Pronominal Anaphora Resolution*. Computational Linguistics, 20(4):535–561.

[LEUNG und KAN 1997] LEUNG, C.H. und W. KAN (1997). *A Statistical Learning Approach to Automatic Indexing of Controlled Index Terms*. Journal of American Society for Information Science, 48(1):55–66.

[LEWIS 1995] LEWIS, D. (1995). *Evaluating and Optimizing Autonomous Text Classification Systems*. In: *Proceedings of SIGIR 95*, S. 246–254.

[LEWIS 1992] LEWIS, DAVID D. (1992). *An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task*. In: *Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, S. 37–50, Kopenhagen.

[Lin 1995] Lin, Chin-Yew (1995). *Topic Identification by Concept Generalization.* In: *Proceedings of the Thirty-third Conference of the Association of Computational Linguistics (ACL-95)*, S. 308–310, Boston, MA.

[Lin und Hovy 1997] Lin, C.Y. und E. Hovy (1997). *Identifying Topics by Position.* In: *Proceedings of the Applied Natural Language Processing Conference.*

[Lovins 1968] Lovins, Janet B. (1968). *Development of a Stemming Algorithm.* Mechanical Translation and Computational Linguistics, 11(1-2):22–31.

[Mani und Bloedorn 1998] Mani, Inderjeet und E. Bloedorn (1998). *Machine Learning of Generic and User-Focused Summarization.* In: *Proceedings of AAAI'98*, Madison, Wisconsin.

[Marcu 1998] Marcu, D. (1998). *Improving Summarization Through Rhetorical Parsing Tuning.* In: *Proceedings of the Workshop on Very Large Corpora*, Montreal, Canada.

[McElligott und Sorensen 1994] McElligott, Michael und H. Sorensen (1994). *An Evolutionary Connectionist Approach to Personal Information Filtering.* In: *INNC 94 (Fourth Irish Neural Network Conference)*, S. 141–146.

[McKeown und Radev 1995] McKeown, K.R. und D. Radev (1995). *Generating Summaries of Multiple News Articles.* In: *Proceedings of the Eighteenth Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, S. 74–82, Seattle, Washington.

[Mertens 1997] Mertens, Andreas (1997). *Robustes Parsing mit Wortagenten.* In: *Proceedings der 10. GLDV Jahrestagung*, Leipzig.

[Miller 1995] Miller, George A. (1995). *WordNet: A Lexical Database for English.* Communications of the ACM, 38(11):39–41.

[Mitchell 1997] Mitchell, Tom M. (1997). *Machine Learning.* McGraw Hill, New York.

[Mitra et al. 1997] Mitra, M., A. Singhal und C. Buckley (1997). *Automatic Text Summarization by Paragraph Extraction.* In: Mani, I. und M. Maybury, Hrsg.: *Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization*, Madrid, Spanien.

[Morris und Hirst 1991] Morris, Jane und G. Hirst (1991). *Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text.* Computational Linguistics, 17(1):21–48.

[Munoz et al. 1999] Munoz, M., V. Punyakanok, D. Roth und D. Zimak (1999). *A Learning Approach to Shallow Parsing.* In: *Proceedings of EMNLP-VLC'99.*

[Muslea 1999] Muslea, Ion (1999). *Extraction Patterns for Information Extraction Tasks: A Survey.* In: *AAAI'99 Workshop Machine Learning for Information Extraction.* AAAI Press.

[Nakagawa 1997] Nakagawa, H. (1997). *Extraction of Index Words from Manuals.* In: *RIAO 97 Conference Proceedings: Computer-Assisted Information Searching on Internet*, S. 598–611, Montreal, Canada.

[Neumann et al. 1997] Neumann, G., R. Backofen, J. Baur, M. Becker und C. Braun (1997). *An Information Extraction Core System for Real World German Text Processing.* In: *5th International Conference of Applied Natural Language*, S. 208–215, Washington, USA.

[Neumann et al. 2000] Neumann, G., C. Braun und J. Piskorski (2000). *A Divide-and-Conquer-Strategy for Shallow Parsing of German Free Text.* In: *Proceedings of ANLP-2000*, S. 239–246, Seattle, Washington.

[Oard 1997] Oard, Douglas W. (1997). *The State of the Art in Text Filtering.* User Modeling and User-Adapted Interaction, 7(3):141–178.

[Oguro et al. 2000] Oguro, Rei, K. Ozeki, K. Takagi und Y. Zhang (2000). *An Efficient Algorithm for Japanese Sentence Compaction Based on Phrase Importance and Inter-Phrase Dependency.* In: Sojka, Petr, I. Kopecek und K. Pala, Hrsg.: *Proceedings of TSD (Text, Speech and Dialogue) - Third International Workshop*, Bd. 1902 d. Reihe *Lecture Notes in Computer Science*, S. 103–108. Springer.

[Ohtake und Masuyama 2001] Ohtake, Kiyonori und S. Masuyama (2001). *Elimination of Multiple Modifiers in Summarization.* In: *Proceedings of ICCPOL*, S. 282–285.

[Porter 1980] Porter, M. F. (1980). *An Algorithm for Suffix Stripping.* Program, 14(3):130–137.

[Quinlan 1993] Quinlan, John Ross (1993). *C4.5: Programs for Machine Learning.* Machine Learning. Morgan Kaufmann, San Mateo, CA.

[Ramshaw und Marcus 1995] Ramshaw, Lance und M. Marcus (1995). *Text Chunking Using Transformation-Based Learning.* In: Yarovsky, David und K. Church, Hrsg.: *Proceedings of the Third Workshop on Very Large Corpora*, S. 82–94, Somerset, New Jersey. Association for Computational Linguistics.

[Rath et al. 1961] Rath, G.J., A. Resnick und T. Savage (1961). *The Formation of Abstracts by the Selection of Sentences.* American Documentation, 12(2):139–143.

[Riloff 1993] Riloff, Ellen (1993). *A Corpus-Based Approach to Domain-Specific Text Summarisation: A Proposal.* In: Endres-Niggemeyer, B., J. Hobbs und K. Sparck-Jones, Hrsg.: *Workshop on Summarising Text for Intelligent Communication*, Dagstuhl, BRD.

[Riloff 1997] Riloff, Ellen (1997). *Little Words Can Make a Big Difference for Text Classification.* In: *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, S. 130–136.

[Riloff und Jones 1999] Riloff, Ellen und R. Jones (1999). *Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping*. In: *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, Orlando, Florida.

[Riloff und Lehnert 1994] Riloff, Ellen und W. Lehnert (1994). *Information Extraction as a Basis for High-Precision Text Classification*. ACM Transactions on Information Systems, 12(3):296–333.

[Rüping 2000] Rüping, Stefan (2000). *mySVM-Manual*. Universität Dortmund, Lehrstuhl Informatik VIII. http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/.

[Salton und Buckley 1988] Salton, G. und C. Buckley (1988). *Term Weighting Approaches in Automatic Text Retrieval*. Information Processing and Management, 24(5):513–523.

[Salton 1989] Salton, Gerard (1989). *Automated Text Processing*. Addison-Wesley.

[Sanderson 1998] Sanderson, Mark (1998). *Accurate User Directed Summarization from Existing Tools*. In: *Proceedings of the 7th International Conference on Information and Knowledge Management (CIKM 98)*, S. 45–51.

[Schewe 1997] Schewe, Sandra (1997). *Automatische Kategorisierung von Volltexten unter Anwendung von NLP-Techniken*. Diplomarbeit, Fachbereich Informatik, Universität Dortmund.

[Shieber 1987] Shieber, Stuart M. (1987). *Evidence Against the Context-Freeness of Natural Language*. In: Savitch, W. J., E. Bach, W. Marsh und G. Safran-Naveh, Hrsg.: *The Formal Complexity of Natural Language*, S. 320–334. Reidel, Dordrecht.

[Sparck-Jones und Galliers 1996] Sparck-Jones, Karen und J. R. Galliers (1996). *Evaluating Natural Language Processing Systems: An Analysis and Review*. Springer Verlag Inc., New York, NY, USA.

[Steinbach et al. 2000] Steinbach, Michael, G. Karypis und V. Kumar (2000). *A Comparison of Document Clustering Techniques*. In: *TextMining Workshop, KDD*.

[Strzalkowski et al. 1998] Strzalkowski, T., J. Wang und B. Wise (1998). *A Robust Practical Text Summarization System*. In: *AAAI Intelligent Text Summarization Workshop*, S. 26–30, Stanford, CA.

[Tarvainen 1981] Tarvainen, Kalevi (1981). *Einführung in die Dependenzgrammatik*. Niemeyer, Tübingen.

[Tombros und Sanderson 1998] Tombros, A. und M. Sanderson (1998). *Advantages of Query Biased Summaries in Information Retrieval*. In: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, S. 2–10.

[TURNEY 2000] TURNEY, PETER D. (2000). *Learning Algorithms for Keyphrase Extraction*. Information Retrieval, 2(4):303–336.

[VAPNIK 1982] VAPNIK, V. (1982). *Estimation of Dependencies Based on Empirical Data*. Springer.

[VAPNIK 1998] VAPNIK, V. (1998). *Statistical Learning Theory*. Wiley, Chichester, GB.

[VELTMANN 1997] VELTMANN, GEORG (1997). *Einsatz eines Multiagentensystems zur Erstellung eines persönlichen Pressespiegels*. Diplomarbeit, Fachbereich Informatik, Universität Dortmund, Germany.

[WAHLSTER 1993] WAHLSTER, W. (1993). *Verbmobil: Translation of Face-to-Face Dialogues*. In: *Proceedings of the Third European Conference on Speech Communication and Technology*, Berlin.

[WEGENER 1993] WEGENER, INGO (1993). *Theoretische Informatik*. Teubner, Stuttgart.

[YANG und LIU 1999] YANG, Y. und X. LIU (1999). *A Re-examination of Text Categorization Methods*. In: *22nd Annual International SIGIR*, S. 42–49, Berkley.

[YANG 1999] YANG, YIMING (1999). *An Evaluation of Statistical Approaches to Text Categorization*. Journal of Information Retrieval, 1(1-2):69–90.

[YANG und PEDERSEN 1997] YANG, YIMING und J. O. PEDERSEN (1997). *A Comparative Study on Feature Selection in Text Categorization*. In: *Proceedings of 14th International Conference on Machine Learning*, S. 412–420. Morgan Kaufmann.

[ZAMIR und ETZIONI 1998] ZAMIR, OREN und O. ETZIONI (1998). *Web Document Clustering: A Feasibility Demonstration*. In: *Proceedings of ACM SIGIR'98*, S. 46–54.

[ZECHNER 1996] ZECHNER, K. (1996). *Fast Generation of Abstracts from General Domain Text Corpora by Extracting Relevant Sentences*. In: *Proceedings of the Sixteenth International Conference on Computational Linguistics*, S. 986–989.

Index