Marcus Hassler

# Neuronale Netze und Computerlinguistik für Information Retrieval

## Eine experimentelle Evaluation

# DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

Angewandte Informatik

Universität Klagenfurt
Fakultät für Wirtschaftswissenschaften und Informatik

Begutachter: O.Univ.Prof. Dipl.-Ing. Mag. Roland Mittermeir
Institut: Institut für Informatik-Systeme (ISYS)

Oktober/2003

## Ehrenwörtliche Erklärung

Ich erkläre ehrenwörtlich, dass ich die vorliegende Schrift verfasst und die mit ihr unmittelbar verbundenen Arbeiten selbst durchgeführt habe. Die in der Schrift verwendete Literatur sowie das Ausmaß der mir im gesamten Arbeitsvorgang gewährten Unterstützung sind ausnahmslos angegeben. Die Schrift ist noch keiner anderen Prüfungsbehörde vorgelegt worden.

.....

Klagenfurt, 23 Oktober 2003

# Zusammenfassung

Im heutigen Umfeld des Internets und der elektronischen Datenverwaltung ist die Organisation von Informationen ein wichtiges Thema. Ein Beschäftigungsfeld der Informatik, das sich diesem Thema widmet, ist das Information Retrieval. Ein Teilgebiet dieses Aufgabenfeldes beschäftigt sich mit der Gruppierung von Textdokumenten, deren Ziel ein Zusammenfassen ähnlicher Texte in denselben Gruppen ist. Diese Sortierung vereinfacht Aufgaben wie die Suche oder das Katalogisieren beträchtlich.

Um Textgruppierungen durchzuführen, sind Verfahren aus dem Bereich des Natural Language Processings einsetzbar, welche relevante Wörter aus den Texten extrahieren. Diese Indexterme dienen der Repräsentation des Inhalt eines Textes. Während der Gruppierung werden diese Repräsentationen verwendet, um Dokumentgruppen zu generieren. Jede dieser Gruppen wird durch einen Repräsentanten dargestellt. Diesen Vorgang übernehmen in dieser Arbeit Neuronale Netze. Während des Retrievals wird eine Abfrage mit allen Gruppenrepräsentanten verglichen, wobei ähnliche Gruppen als für den Benutzer relevant erachtet werden.

Im Zuge dieser Arbeit wurde der für englischsprachige Texte konzipierte Prototyp SyRS auf deutschsprachige Texte adaptiert. Die Repräsentation in SyRS basierte ürsprünglich auf der Thema-Rhema Theorie. Zusätzlich wurde eine andere Repräsentationsform mittels des Vektorenmodells implementiert, welche dem Thema-Rhema Modell gegenübergestellt wurde. Weiters wurden jeweils zwei linguistische Varianten zu diesen Modellen entwickelt: Eine light-Variante bezog nur Nomen und Verben in die Analyse ein, während eine heavy-Variante Nomen, Verben, Adjektive und Adverbien berücksichtigte. Alle vier Varianten kamen bei zwei unterschiedlichen Subsets eines Textcorpus zum Einsatz, um sie miteinander zu vergleichen.

Ein Vergleich der linguistischen Varianten ergab in beiden Corpora keine nennenswerten Leistungsunterschiede. Die errechneten Kennzahlen beider Modelle zeigten nur unbedeutende Abweichungen zugunsten des Vektorenmodells. Aufgrund der ähnlichen Ergebnisse, jedoch einer geringeren Trainingsdauer und einer niedrigeren Komplexität, ist das Vektorenmodell eindeutig zu favorisieren.

## Abstract

Today, we are observing an ever-growing amount of electronic information due to technical advances (e.g. internet). The challange is then the organization of this informations in a way to make it easily accessible to seekers/users. Information Retrieval is a field of computer science that is concerned with organizing and searching of off- and online documents. Document organization means grouping similar documents under a common topic. The aim of doing so is to reduce the search time for relevant documents to a given user's query. To achieve this goal, natural language techniques are applied to identify relevant terms, called indexing terms, that reflect the semantic content of the documents. The set of index terms will serve as representation. The process of grouping uses the resulting representations to partition the documents into coherent groups. Each group is represented by its prototype (representative). The present work applies neural networks to perform the clustering process. During retrieval, the user's query is compared to the prototypes only. Matching prototypes indicate relevant clusters (topics) to the user.

In this work, an existing software prototype called SyRS (Systemic Retrieval System), tailored for English texts was adapted for German. The original version of SyRS relied on the theme-rheme theory to represent documents. For an assessment another representation, the vector space model, has been implemented. These two representations were compared. Furthermore, two linguistic variants of each representation model were studied and their effects on the accuracy of clustering were analyzed. The first variant considers nouns and verbs only, while the second variant includes nouns, verbs, adjectives and adverbs. To evaluate the four variants, we used two distinct subsets of a preclassified corpus.

The empirical results showed that the linguistic variants applied on both corpora had no major effect on the outcome. Using the same performance measures allowed us to address the comparison between the theme-rheme and the vector space model in a clean way. The experiments illustrated that the vector space model provided slightly better results. It is worth mentioning, that the processing time as well as the complexity of the vector space model are much lower than those of th original Theme-Rheme Model.

# Danksagung

Diese Zeilen möchte ich an einige Personen richten, die mich während der Durchführung dieser Arbeit tatkräftig unterstützt und angespornt haben. Zunächst richtet sich mein Dank an meinen Betreuer, *Dr. Abdelhamid Bouchachia*, der mich durch seine fachliche Kompetenz und kritischen Hinweise von Beginn an geleitet und unterstützt hat. Gleichermaßen gilt mein Dank *Prof. Roland Mittermeir*, der mir trotz seiner begrenzten Zeitressourcen immer wieder seine kostbare Aufmerksamkeit schenkte, um mir mit wertvollen Ratschlägen und einem offenen Ohr zur Seite zu stehen. Besonders in der arbeitsintensiveren Abschlussphase, in welcher Skepsis bezüglich der Termineinhaltung durchaus angebracht war, stand er hinter mir.

Weiters will ich meinen armen und vor allem „freiwilligen" Opfern danken, die ich bat, meine Arbeit Korrektur zu lesen. Dazu zählen neben meiner geschätzten *Magdalena* auch meine Mutter, *Ursula*, und meine Schwester, *Kathrin*. Durch ihre „Entdeckungen" und Vorschläge ist es mir möglich gewesen, die Arbeit auf den nun vorliegenden Stand zu bringen.

Darüber hinaus gilt mein Dank den (Arbeits-)Kollegen der Universität Klagenfurt, die ich in Zeiten der Verwirrung aufsuchen konnte, um nach Rat zu fragen oder einfach nur um abgelenkt zu werden. Sicherlich sind hier die ehrenwerten Sir's der *Hekkas-Gilde* sowie andere „*Leidgenossen*" (Diplomanden), mit denen ich mich Austauschen konnte, zu erwähnen. Alle *weiteren Personen*, die bisher nicht direkt angesprochen wurden, sich aber gerne an dieser Stelle wiederfinden würden, seien hiermit ebenfalls bedacht...

Weitere Anerkennung gebührt den *Bediensteten meines Stammlokals* - die nebst geistiger auch die hin und wieder notwendige, entsprechende flüssige Nahrung zur Verfügung stellten - und bei der Einhaltung der Sperrstunde die oftmals erforderliche Grosszügigkeit walten liesen.

Und zu guter Letzt gebührt mein aufrichtiger Dank meinen *Eltern*, für den wohl wichtichsten Beitrag zum Enstehen dieser Arbeit, in Form meiner Geburt vor nunmehr 26 Jahren. *zwinker*

# Abkürzungsverzeichnis

| | |
|---|---|
| ***A*** | Accuracy |
| ***ART*** | Adaptive Resonance Theory |
| ***BEP*** | Breakeven Point |
| ***BIN*** | Bayesian Inference Network |
| ***DR*** | Data Retrieval |
| ***DC*** | Dokumentclustering |
| ***DK*** | Dokumentkategorisierung |
| ***E*** | Error |
| ***F*** | Fallout |
| ***FAM*** | Fuzzy Associative Memory |
| ***GDR*** | globale Dimensionsreduktion |
| ***idf*** | inverse document frequency |
| ***IR*** | Information Retrieval |
| ***kNN*** | k Nearest Neighbor |
| ***LDR*** | lokalen Dimensionsreduktion |
| ***LSI*** | Latent Semantic Indexing |
| ***NLP*** | Natural Language Processing |
| ***NN*** | Neuronales Netz |

| | |
|---|---|
| ***O*** | Overlap |
| ***P*** | Precision |
| ***POS*** | Part-of-Speech |
| ***PRP*** | Probability Ranking Principle |
| ***R*** | Recall |
| ***SDI*** | Selective Dissemination of Information |
| ***SFT*** | Systemic Functional Theory |
| ***SOM*** | Self-Organizing Maps |
| ***STTS*** | Stuttgart-Tübigen Tagset |
| ***SyRS*** | Systemic Retrieval System |
| ***VSM*** | Vector Space Model, Vektorenmodell |
| ***SVM*** | Support Vector Machines |
| ***tf*** | term frequency |
| ***Th-Rh*** | Thema-Rhema Modell |
| ***TR*** | Textretrieval |
| ***TF*** | Textfilterung |
| ***WSD*** | Word Sense Disambiguation |

# Inhaltsverzeichnis

**1 Einleitung** **1**

1.1 Problemstellung . . . . . 1

1.2 Ziel der Arbeit . . . . . 3

1.3 Aufbau der Arbeit . . . . . 4

**2 Information Retrieval** **8**

2.1 Begriffsabgrenzung . . . . . 8

2.2 Der Retrieval Prozess . . . . . 10

2.3 Aufgabenbereiche des Information Retrieval . . . . . 12

2.3.1 Textretrieval . . . . . 12

2.3.2 Textfilterung . . . . . 12

2.3.3 Textgruppierung . . . . . 13

2.4 Repräsentation von Texten . . . . . 15

2.5 Clusterretrieval . . . . . 18

**3 Textgruppierung** **21**

3.1 Dokumentkategorisierung . . . . . 21

3.2 Dokumentclustering . . . . . 24

3.3 Ähnlichkeits- und Abstandsmaße . . . . . 27

3.4 Modelle zur Dokumentgruppierung . . . . . 28

3.4.1 Statistische Modelle . . . . . 28

3.4.2 Probabilistische Modelle . . . . . 29

3.4.3 Genetische Algorithmen . . . . . 31

3.4.4 Clustering Algorithmen . . . . . 34

3.4.5 Neuronale Netze . . . . . 35
3.5 Evaluation von Dokumentgruppierungen . . . . . 39

**4 Dokumentrepräsentation 41**

4.1 Überblick . . . . . 41
4.2 Methoden der Textanalyse . . . . . 46
4.2.1 Grundlagen . . . . . 46
4.2.2 Tokenisierung (Tokenizing) . . . . . 48
4.2.3 Identifikation von Wortkategorien (Tagging) . . . . . 50
4.2.4 Wortnormalisierung (Stemming) . . . . . 51
4.2.5 Filterung (Stopwortlisten) . . . . . 53
4.3 Gewichtung mittels des Vektorenmodells . . . . . 54
4.4 Gewichtung mittels des Thema-Rhema Modells . . . . . 60

**5 Das SyRS System - Der SyRS Prototyp 71**

5.1 Architektur . . . . . 71
5.2 Das Natural Language Modul . . . . . 72
5.2.1 Aufbau . . . . . 72
5.2.2 Der Tokenizer . . . . . 75
5.2.3 Der Tagger . . . . . 76
5.2.4 Der Stemmer . . . . . 79
5.2.5 Die Stopwortliste . . . . . 80
5.3 Die Gewichtung . . . . . 80
5.3.1 Aufbau . . . . . 80
5.3.2 Das Vektorenmodell . . . . . 82
5.3.3 Das Thema-Rhema Modell . . . . . 82
5.4 Das Neuronale Netzwerk Modul . . . . . 86

5.4.1 Aufbau . . . . . 86
5.4.2 Das Fuzzy Associative Memory (FAM) Netz . . . . . 88
5.4.3 Das fuzzy Adaptive Resonance Theory (ART) Netz . . . . . 91
5.4.4 Zusammenführung von FAM und ART . . . . . 95

**6 Evaluation von SyRS** **97**

6.1 Einleitung . . . . . 97
6.2 Evaluationsmetriken . . . . . 99
6.2.1 Precision und Recall . . . . . 99
6.2.2 Alternative Kennzahlen . . . . . 102
6.3 Das Corpus . . . . . 105
6.4 Ergebnisse der Experimente . . . . . 107
6.4.1 Experimentelle Methodik . . . . . 107
6.4.2 Parametereinstellungen . . . . . 109
6.4.3 Das Training . . . . . 110
6.4.4 Das Testen . . . . . 117
6.4.5 Vergleich: Vektorenmodell vs. Thema-Rhema Modell . . . . . 128
6.4.6 Gewonnene Erkenntnisse . . . . . 135
6.5 Dokumentclustering und Information Retrieval . . . . . 136

**7 Resümee** **139**

7.1 Zusammenfassung . . . . . 139
7.2 Erweiterungspotential von SyRS . . . . . 143
7.2.1 Das Natural Language Modul . . . . . 143
7.2.2 Das Neuronalen Netzwerk Modul . . . . . 146
7.3 Ausblick . . . . . 148

**A Einführung in Fuzzy-Sets** **149**

**Literaturverzeichnis** **153**

# Abbildungsverzeichnis


1.1 Aufbau der Arbeit . . . . . 6

2.1 Grundprinzip des Information Retrievals . . . . . 9

2.2 Textfilterung und Textretrieval . . . . . 14

2.3 Globales versus abfragebasiertes Clustering . . . . . 15

2.4 Textrepräsentation - Vom Volltext zu Indextermen . . . . . 17

2.5 Clusterretrieval . . . . . 19

3.1 Dokumentkategorisierung neuer Dokumente . . . . . 23

3.2 Arbeitsweise von Genetischen Algorithmen . . . . . 32

3.3 Dokumentclustering mit genetischen Algorithmen . . . . . 33

3.4 Neuronales Netz zur Dokumentkategorisierung . . . . . 37

4.1 Verteilung sortierter Wortfrequenzen (links) und Größe des Vokabulars (rechts) . . . . . 45

4.2 Textrepräsentation - Vom Volltext zu Indextermen . . . . . 48

4.3 Ähnlichkeitsberechnung . . . . . 55

4.4 Dokumentrepräsentation des Thema-Rhema Modells . . . . . 70

5.1 SyRS Übersichtsgrafik . . . . . 72

5.2 SyRS - Das Natural Language Modul . . . . . 73

5.3 Entscheidungsbaum . . . . . 77

5.4 Die Thema-Rhema Analyse . . . . . 84

5.5 Version 2: Thema-Rhema Matrix . . . . . 86

5.6 SyRS - Das Neuronale Netzwerk Modul . . . . . 86

5.7 Das Fuzzy Associative Memory Netz (FAM) . . . . . 88

5.8 Mappingalgorithmus des FAM . . . . . 90
5.9 Das Adaptive Resonance Theory Netz (ART) . . . . . 92
5.10 Clusteralgorithmus des fuzzy ART . . . . . 94
5.11 Rhema-Thema Mapping (FAM) und Clustering (ART) . . . . . 96

6.1 Clustergröße - Vergleich Corpus 1 . . . . . 116
6.2 Clustergröße - Vergleich Corpus 2 . . . . . 116
6.3 VSM light, Corpus 1 . . . . . 118
6.4 VSM heavy, Corpus 1 . . . . . 118
6.5 VSM light vs. heavy, Corpus 1 . . . . . 118
6.6 VSM light, Corpus 2 . . . . . 121
6.7 VSM heavy, Corpus 2 . . . . . 121
6.8 VSM light vs. heavy, Corpus 2 . . . . . 121
6.9 Th-Rh light, Corpus 1 . . . . . 124
6.10 Th-Rh heavy, Corpus 1 . . . . . 124
6.11 Th-Rh light vs. heavy, Corpus 1 . . . . . 124
6.12 Th-Rh light, Corpus 2 . . . . . 127
6.13 Th-Rh heavy, Corpus 2 . . . . . 127
6.14 Th-Rh light vs. heavy, Corpus 2 . . . . . 127
6.15 VSM light vs. Th-Rh light, Corpus 1 . . . . . 130
6.16 VSM heavy vs. Th-Rh heavy, Corpus 1 . . . . . 130
6.17 VSM vs. Th-Rh, F-Measure, Corpus 1 . . . . . 130
6.18 VSM light vs. Th-Rh light, Corpus 2 . . . . . 133
6.19 VSM heavy vs. Th-Rh heavy, Corpus 2 . . . . . 133
6.20 VSM vs. Th-Rh, F-Measure, Corpus 2 . . . . . 133

7.1 Verbessertes SyRS . . . . . 147

# Tabellenverzeichnis


2.1 Information Retrieval versus Data Retrieval . . . . . 10

3.1 Kategorisierungsmatrix . . . . . 22

4.1 Thema-Rhema Beispiel . . . . . 67

4.2 Thema-Rhema Matrix . . . . . 67

5.1 Verwendetes POS-Tagset . . . . . 78

5.2 „light“und „heavy“Variante . . . . . 82

6.1 Kontingenztabelle (Multiple Binary Classification) . . . . . 99

6.2 Übersicht über die gebildeten Corpora . . . . . 107

6.3 Analyseergebnisse der Corpora . . . . . 110

6.4 Trainingsergebnisse VSM (Corpus 1 & Corpus 2) . . . . . 112

6.5 Trainingsergebnisse Th-Rh (Corpus 1 & Corpus 2) . . . . . 114

6.6 Testergebnis VSM, Corpus 1 . . . . . 117

6.7 Testergebnis SVM, Corpus 2 . . . . . 120

6.8 Testergebnis Th-Rh, Corpus 1 . . . . . 123

6.9 Testergebnis Th-Rh, Corpus 2 . . . . . 126

6.10 Testergebnisse - Breakeven Point Analyse . . . . . 128

6.11 Testergebnisse - F-Measure . . . . . 129

6.12 Beispiel eines Abfrageergebnisses (1) . . . . . 137

6.13 Beispiel eines Abfrageergebnisses (2) . . . . . 137

KAPITEL 1

# Einleitung

## 1.1 Problemstellung

Im Zeitalter des Internets steht jedem Benutzer[1] eine wahre Flut an Informationen zur Verfügung. Diese schnell, billig und einfach zugänglichen Informationen liegen jedoch oft in unstrukturierter und heterogener Form, unter mangelhaften oder gar falschen Namen, unzureichend beschrieben oder unauffindbar vor. All diese Punkte machen es oft schwierig, die gewünschten Informationen zu beziehen und/oder diese in geeigneter Form zu organisieren [15].

Neben multimedialen Inhalten stehen diese Informationen meist in Form von schriftlichen, natürlichsprachlichen Texten zur Verfügung. Gerade die natürliche Sprache erlaubt es, komplizierte Sachverhalte und Meinungen auszudrücken. Ein grammatikalisches Regelwerk ermöglicht dabei eine Fokussierung verschiedener Elemente, natürlich innerhalb gewisser Grenzen. Bei der semantischen Interpretation von Texten treten dadurch jedoch Probleme auf [95]:

1. Der Autor des Textes formuliert den Inhalt nicht genau oder nur vage, da die natürliche Sprache großen Spielraum für Interpretationen einräumt.

2. Verschiedene Wörter können dieselbe Bedeutung haben und somit dasselbe auf verschiedene Weise beschreiben.

3. Dieselben Wörter können unterschiedliche Bedeutungen in verschiedenen Kontexten ausdrücken.

[1] Stellvertretend für den weiteren Verlauf der Arbeit wird an dieser Stelle darauf hingewiesen, dass die weibliche Form der jeweiligen Substantiva nicht separat erwähnt wrd.

Heutzutage stößt man nahezu in allen Bereichen der Informationsverarbeitung auf sehr große Datenmengen, welche sich über die Jahre angesammelt haben. Einiges davon wurde unter Umständen auch (semi-)automatisch aus dem Internet bezogen. Anderes wurde von einer Vielzahl an Autoren mit unterschiedlichen Ablagesystemen und Intensionen produziert. Diese Dokumente liegen oftmals nur unstrukturiert vor und folgen keinem einheitlichen Standard. Eines der daraus resultierenden Hauptprobleme ist das nicht Wiederauffinden von gesuchten Informationen in diesen umfangreichen Datenpools. Ohne vernünftige Möglichkeiten zur Organisation und Suche innerhalb dieser Datensammlungen sind Informationen daher oftmals unauffindbar verloren („Datenfriedhof“).

Aus diesem Grund haben viele Forscher begonnen, Verfahren zur Verbesserung solcher Informationssuchen zu entwickeln. Dieses Aufgabenfeld ist unter dem Namen Information Retrieval (siehe Kapitel 2) bekannt geworden, das verschiedene Ansätze zur Lösung dieser Problematik bereitstellt [90].

Ein möglicher Ansatzpunkt ist die Aufteilung aller zur Verfügung stehenden Daten in kleinere, überschaubare Teile. Dabei wird versucht, Dokumente nach bestimmten gemeinsamen Eigenschaften zu gruppieren, sodass ähnliche Dokumente derselben Gruppe angehören. Auf diese Art kann die Anzahl der zu untersuchenden Dokumente (der Suchraum) beträchtlich verkleinert werden, wodurch schnellere und geeignetere Ergebnisse bei der Suche erzielt werden können. Eine solche Vorauswahl kann in einem nächsten Schritt als Ausgangspunkt einer näheren Untersuchung zur Bestimmung der Relevanz [59] der einzelnen Dokumente dienen.

Besonders in den Bereichen der automatisierten Dokumentkategorisierung und dem Dokumentclustering wurde die Forschung in den letzten Jahren intensiviert [32, 51]. Als Ergebnis findet man heute viele Methoden und Systeme im Einsatz, die sich dieser Aufgaben annehmen. Jedes dieser Verfahren hat spezifische Stärken und Schwächen, die je nach Anwendungsgebiet, Zeitaufwand, Texttypus, usw. variieren. Somit müssen für bestimmte Aufgabenfelder mehrere Verfahren herangezogen, ausgetestet und evaluiert werden, wodurch Vergleichswerte sehr wertvoll werden. Die Forschungen an solchen Verfahren sind jedoch noch lange nicht abgeschlossen.

## 1.2 Ziel der Arbeit

Die vorliegende Arbeit stellt einen Überblick über derzeitige Möglichkeiten und Einsatzgebiete von Information Retrieval Systemen dar. Hierbei soll im Speziellen auf das Anwendungsgebiet der Textgruppierung eingegangen werden.

In diesem Zusammenhang soll ein grundlegendes Verständnis über die Problematik der inhaltlichen Repräsentation von (deutschen) Textdokumenten vermittelt werden. Deshalb ist es notwendig auf Lösungsansätze aus dem Bereich des Natural Language Processings (NLP) zurückzugreifen. Hierbei werden besonders die Einsatzmöglichkeiten der Thema-Rhema Theorie zur Textanalyse und -repräsentation deutscher Texte aufgezeigt.

Im Zuge dieser Arbeit wird ein bestehender Prototyp für die Aufgabe des Textclusterings deutscher Texte adaptiert. Da dieser Prototyp auf Neuronalen Netzen beruht, wird der Leser ebenfalls in die Arbeits- und Funktionsweise dieser eingeführt.

Ziel der Arbeit ist die Evaluation des adaptierten Prototyps. Deshalb müssen Metriken zur Evaluation vorgestellt und definiert werden. Die Grundlage einer Evaluation solcher Systeme stellt ein vorkategorisiertes Corpus dar, anhand dessen die Effektivität des Systems ermittelt wird.

Um die Evaluationsergebnisse angemessen interpretieren zu können, ist ein Vergleichsmodell notwendig. Hierbei soll ein Standardmodell, das Vektorenmodell, dienen, welches unter denselben Bedingungen evaluiert wird. Anschliessend sollen die ermittelten Evaluationsmetriken einander gegenübergestellt und kritisch hinterfragt werden.

Um das Ziel der Arbeit zu erreichen, sind folgende Teilaufgaben durchzuführen:

- Eine geeignete Repräsentation der Dokumente muss in beiden Modellen, dem Thema-Rhema Modell und dem Vektorenmodell, gewählt werden. Die Textdokumente müssen entsprechend formatiert und für weitere Verarbeitungsschritte aufbereitet werden. Bei der Textanalyse kommen hier Methoden des NLP zum Einsatz, die den Text des Dokuments in eine geeignete Repräsentationsform transformieren.

- Ein Vergleichsmechanismus zwischen zwei Dokumentrepräsentationen muss vorhanden sein. Diese Vergleichsfunktion muss das Ergebnis in messbaren Zahlenwerten darstellen, um den Grad an Übereinstimmung objektiv feststellen zu können.

- Zwei verschiedene Modelle zur Dokumentrepräsentation sollen verglichen werden. Einerseits soll der bereits bestehende Prototyp basierend auf der Thema-Rhema Theorie verwendet werden. Zusätzlich soll ein Standard-Vergleichsmodell, das Vektorenmodell, implementiert werden, um die Ergebnisse des Prototyps beurteilen zu können.

- Ein Corpus ist notwendig, um die beiden Modelle zu evaluieren. Da nur wenige deutschsprachige Textcorpora frei verfügbar sind, muss ein eigenes Corpus selbstständig erstellt werden.

- Geeignete Evaluationsmetriken müssen für die Bewertung des Clusteringprozesses ausgewählt und definiert werden. Hierbei sollen Standardmetriken wie Recall und Precision, wie sie bei der Evaluation solcher Systeme verwendet werden, zum Einsatz kommen.

- Die Evaluationsergebnisse müssen einander gegenübergestellt und miteinander verglichen werden. Es soll eine Aussage über die Effektivität der Systeme getroffen werden können. Der Vergleich der Systeme untereinander spiegelt den Beitrag der Thema-Rhema Theorie bei der Dokumentrepräsentation im Verhältnis zu anderen Modellen wider.

## 1.3 Aufbau der Arbeit

Ihren Ausgangspunkt nimmt die Arbeit in Anlehnung an das umfassende Thema des Information Retrieval. Es wird auf grundlegende Ansätze eingegangen, der Retrievalprozess genauer betrachtet und einige Aufgabenbereiche vorgestellt. Ein spezielles Anwendungsfeld des Information Retrieval auf dem diese Arbeit beruht, die Textgruppierung, wird im Speziellen erörtert. Anschließend werden Möglichkeiten und Aufgaben der Textrepräsentation behandelt. In diesem Zusammenhang werden zwei

Modelle, das Vektorenmodell und ein Modell basierend auf der Thema-Rhema Theorie, beschrieben, implementiert und einander gegenübergestellt. Den Abschluss der Arbeit bildet ein Ausblick auf zukünftige Forschungsmöglichkeiten und Ansatzpunkte zur Verbesserung des bestehenden Systems zusammen mit einem kurzen Resümee der Darstellung.

Die Basis der Arbeit stellt der von Dr. Bouchachia im Zuge seiner Dissertation entwickelte Prototyp SyRS dar. SyRS (**Sy**stemic **R**etrieval **S**ystem) ist ein natürlichsprachliches Information Retrieval System für englischsprachige Texte. Im Zuge einer Projektmitarbeit wurde das System vom Autor dieser Arbeit für das Deutsche reimplementiert.

Zur Evaluation dieses Systems wurde ein Corpus, basierend auf der Diplom- und Dissertationsdatenbank Diplomica[2], erstellt. Nach einer Adaption von SyRS an das Corpus wurde das System mit verschiedenen Parametereinstellungen getestet und Evaluationsmetriken errechnet. Um diese Werte mit anderen Systemen vergleichen zu können, wurde ein Standardmodell, das Vektorenmodell, implementiert und mit demselben Corpus und gleichen Parametereinstellungen getestet und evaluiert. Im Anschluss daran wurden die ermittelten Evaluationsergebnisse beider Modelle miteinander verglichen.

Diese Arbeit konzentriert sich auf den Aspekt des Dokumentclusterings und ist in sieben Kapiteln organisiert. Einen Überblick über die Inhalte und Zusammenhänge der einzelnen Kapitel gibt Abbildung 1.1.

Kapitel 2 gibt einen Überblick über Information Retrieval und stellt einige Aufgabengebiete vor.

Im Kapitel 3 wird das Thema der Textgruppierung genauer beleuchtet. Es wird auf die Grundlagen des Kategorisierens und Clusterings eingegangen sowie verschiedene Modelle für ein automatisches Clustering beschrieben.

Kapitel 4 befasst sich anschließend mit der Dokumentrepräsentation. Es werden die einzelnen Schritte der Text- und Dokumentanalyse vorgestellt und erörtert. Hierbei liegt der Fokus auf zwei zugrundeliegenden Modellen, dem Vektorenmodell und dem

[2] Zu finden unter http://www.diplomica.com (Stand: 03.03.2003).

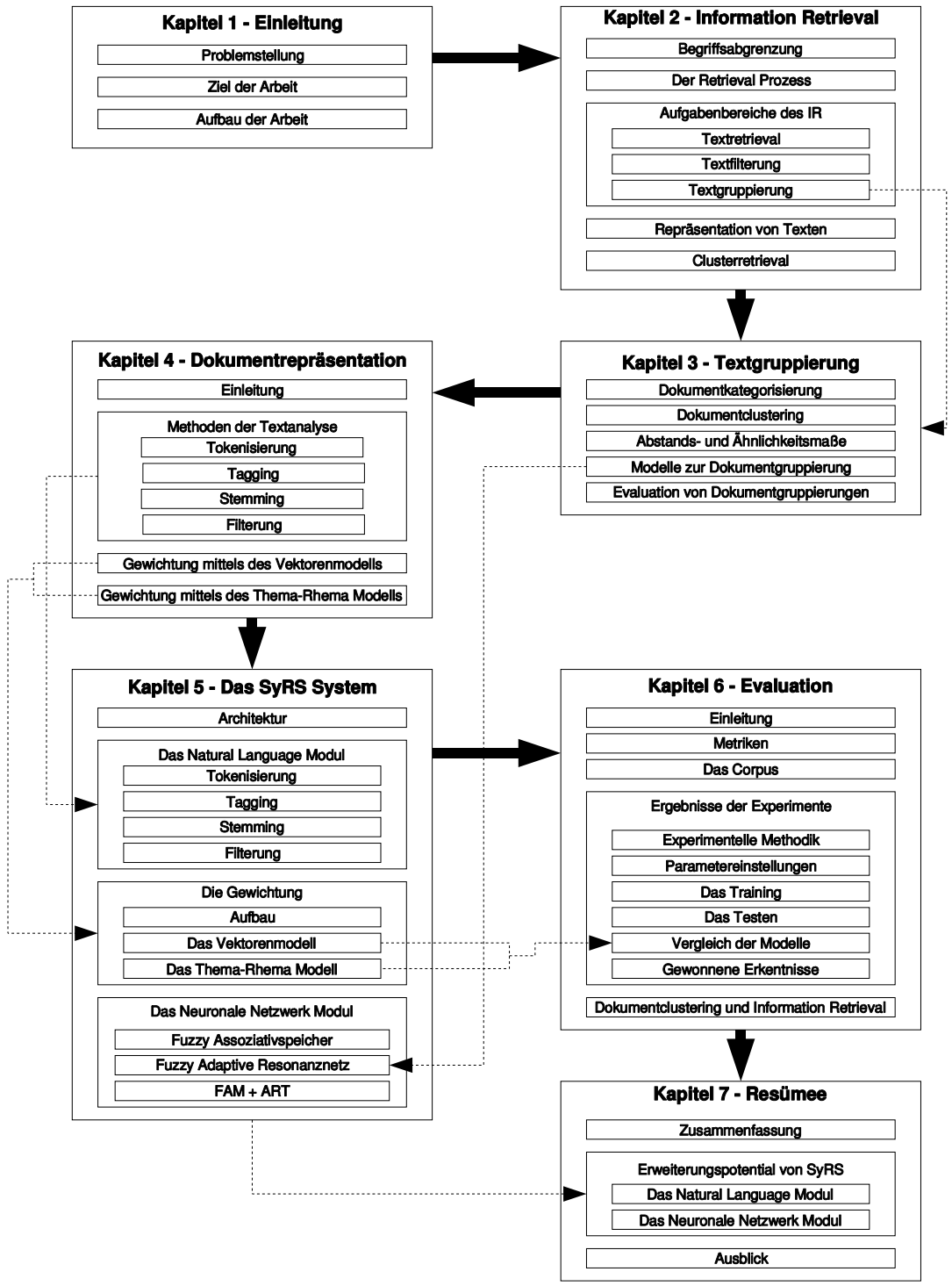



Abbildung 1.1: Aufbau der Arbeit

Thema-Rhema Modell der Systemic Functional Grammar, die ebenfalls die Basis für den bereits vorliegenden Prototypen bilden.

Im Kapitel 5 wird der entwickelte Prototyp vorgestellt. Der Aufbau des Systems wird zusammen mit Erklärungen zu den einzelnen Komponenten im Detail erläutert.

Das Kapitel 6 beschäftigt sich mit der Bewertung von SyRS und geht auf die Ergebnisse der Evaluation ein. Dabei werden die erzielten Resultate der verschiedenen Modelle präsentiert und anschließend miteinander verglichen.

Das letzte Kapitel 7 schließt diese Arbeit und fasst den vorgestellten Sachverhalt noch einmal zusammen. Darüber hinaus wird ein Ausblick auf die zukünftige Entwicklung des Natural Language Processings im Information Retrieval gegeben und mögliche Anknüpfungspunkte für Verbesserungen und Erweiterungen am SyRS Prototyp gegeben.

KAPITEL 2

# Information Retrieval

Die hier vorliegende Arbeit ist im Anwendungsfeld des Information Retrieval angesiedelt. Dieses Kapitel gibt einen Überblick über diesen Bereich und nimmt eine Begriffsabgrenzung vor. Im Anschluß daran werden einige Aufgabengebiete wie das Textretrieval, die Textfilterung und die Textkategorisierung umrissen. Eine kurze Erläuterung zu Textrepräsentationen sowie ein relativ neues Retrieval Modell, das Clusterretrieval, schließen dieses Kapitel.

## 2.1 Begriffsabgrenzung

Information Retrieval (IR) beschäftigt sich mit der Repräsentation, Speicherung, Organisation von und dem Zugriff auf Informationen [4, 95, 90, 59].

Ein IR System basiert dabei auf drei Konzepten [6]:

1. Einem Modell für die Repräsentation von Dokumenten,
2. einem Modell zur Eingabe von Benutzerabfragen und
3. einem Modell zur Ähnlichkeitsbestimmung zwischen Dokumentrepräsentationen und Benutzerabfragen.

Der eigentliche Vorgang des IR kann in drei Teilaufgaben untergliedert werden. Zuerst werden alle an das System gereichte Daten in geeigneter Form aufbereitet und ihre Repräsentationsform gebildet. Auf diesen Repräsentationen werden anschließend Vergleiche durchgeführt, um Ähnlichkeiten unter ihnen zu finden. Am Ende werden die Dokumente des Ergebnisraumes nach ihrer Ähnlichkeit sortiert und

dem Benutzer entsprechend veranschaulicht. Abbildung 2.1 zeigt das grundsätzliche Vorgehen beim IR.

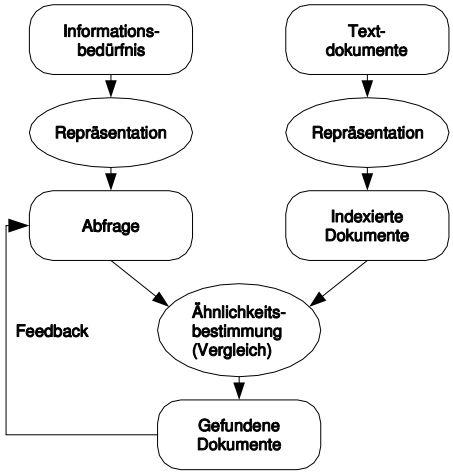



Abbildung 2.1: Grundprinzip des Information Retrievals, aus [6, Seite 6]

Eine geeignete Repräsentation und Organisation von Informationen ermöglicht einen einfachen und schnellen Zugriff auf die gewünschten Daten. Leider ist die Bestimmung der Kriterien einer solchen Repräsentationsform, anhand derer bestimmte Informationen gefunden werden können, nicht trivial [4]. Da Volltextbeschreibungen nur von den wenigsten IR Systemen akzeptiert werden, muss der Benutzer im Regelfall seine Abfrage umformulieren, um sie im System eingeben zu können. In den häufigsten Fällen wird diese Abfrage auf eine Menge von Stichworten abgebildet. In diesem Zusammenhang spricht man von Operationen auf Abfragen (Query Operations) [4]. Ziel eines IR Systems ist es, ausgehend von einer Benutzerabfrage, alle für diese Abfrage relevanten Informationen zurückzuliefern. Eine Definition des Begriffs der „Relevanz“ von Dokumenten ist hierbei ebenfalls nicht trivial, wie Oystein aufzeigt [95].

Die Betonung beim IR liegt auf dem Bezug von Informationen, im Gegensatz zum Data Retrieval (DR). Beim DR handelt es sich vordringlich um das Auffinden von Dokumenten, die bestimmte Schlüsselwörter einer Abfrage enthalten. Als Extremfall kann eine einfache Datenbankabfrage als Beispiel dienen. Die Abfragesprachen basieren auf klar definierten Regeln, wie sie etwa durch Regular Expressions beschrieben werden. Ein einzelnes fehlendes oder fehlerhaftes Objekt innerhalb eines Dokuments bedeutet ein negatives Ergebnis bei der Übereinstimmung mit der Ab-

frage und wird somit als mögliches Resultat der Suche ausgeschlossen. DR baut somit auf vorstrukturierten Informationenquellen auf [4]. Diese Forderung ist beim IR im Fall von natürlichsprachlichen Textdokumenten nur in den seltensten Fällen erfüllt. Beim IR hingegen ist es möglich (und notwendig), ungenaue oder nur teilweise korrekte Ergebnisse ebenso in das Suchresultat mit einzubeziehen. Dies ist ein Hauptgrund für den Einsatz von IR in der Verarbeitung natürlicher Sprache, die oft unstrukturiert vorliegt und semantisch mehrdeutig ist [4]. Van Rijsbergen [88] unterscheidet DR von IR anhand folgender Eigenschaften (siehe Tabelle 2.1).

Tabelle 2.1: Information Retrieval versus Data Retrieval

| **Characteristic** | **Data Retrieval** | **Information Retrieval** |
|---|---|---|
| Matching | Exact Matching | Partial Match, Best Match |
| Inference | Deduction | Induction |
| Model | Deterministic | Probabilistic |
| Classification | Monothetic | Polythetic |
| Query Specification | Complete | Incomplete |
| Items wanted | Matching | Relevant |
| Error Response | Sensitive | Insensitive |

Um die Leistungsfähigkeit von IR Systemen bewerten zu können, wurden unterschiedliche Metriken entwickelt. Die Effektivität wird nicht nur durch die Zeit- und Raumdimension angegeben, sondern kann auch durch andere Kennzahlen wie Recall und/oder Precision ausgedrückt werden, wie sie im Kapitel 6 genauer behandelt werden.

## 2.2 Der Retrieval Prozess

Noch bevor ein (Text-)Retrieval stattfinden kann, muss eine Text- bzw. Dokumentsammlung, ein sogenanntes Corpus, definiert werden. Dazu zählt die Identifikation der zu verwendenden Dokumente, die (Text-)Operationen darauf und das zugrundeliegende Textmodell [4]. Die Textoperationen transformieren die originalen Do-

kumente in ihre Dokumentrepräsentation. Das Textmodell definiert dabei die zugrundeliegende Textstruktur und gibt diejenigen Elemente der Dokumente an, die bezogen werden können. Eine Möglichkeit den Zugriff während des Retrievals zu beschleunigen, besteht darin, einen Index auf die Dokumentrepräsentationen zu legen. Der Aufwand und die Kosten eines solchen Index (Zeit und Speicherkapazität) amortisiert sich schnell bei häufigen Systemabfragen [4].

Der eigentliche Prozess des Beziehens von Dokumenten kann in folgende Einzelschritte unterteilt werden. Zuerst wird eine geeignete Abfrage an das System formuliert. Diese Abfrage wird vom System geparst und denselben Texttransformationen unterworfen wie die originalen Textdokumente des Corpus zuvor. Es ist durchaus möglich zusätzliche Transformationen auf eine Abfrage anzuwenden, um diese systemkonform zu machen. Anschließend wird die Abfrage verarbeitet, indem sie mit allen Dokumenten des Corpus verglichen wird. Ein zuvor erstellter Index führt dabei zu schnelleren Ergebnissen. Bevor die bezogenen Dokumente an den Benutzer zurückgeliefert werden, findet eine Reihung der Ergebnisse statt. Die Dokumente werden dabei ihrem Ähnlichkeitsgrad nach absteigend sortiert. Im Anschluss daran bestimmt der Benutzer manuell die Dokumente, die für ihn von Bedeutung sind. Dieses Benutzerfeedback kann ebenfalls in die Arbeitsweise des Systems integriert werden. Durch diese Informationen ist das System in der Lage, zukünftige Adaptionen an Abfrageformulierungen vorzunehmen und zu erlernen [4]. Ein Überblick über den allgemeinen Ablauf von IR Systemen wurde bereits in Abbildung 2.1 auf Seite 9 gegeben.

Das Feld des Information Retrieval setzt sich mit einer Vielzahl an Aufgabenstellungen auseinander. Eine eindeutige Unterscheidung der verschiedenen Einsatzgebiete ist oftmals nicht möglich. Jedoch beruhen diese in der Regel alle auf der Ähnlichkeitsbestimmung von Repräsentationen (Dokumente, Abfragen, Benutzerprofile, ...). Dadurch eröffnen sich Einsatzgebiete wie Textretrieval, Textfilterung, Textrouting, Dokumentkategorisierung und Dokumentclustering, um nur einige zu nennen [90].

## 2.3 Aufgabenbereiche des Information Retrieval

### 2.3.1 Textretrieval

Um es von anderen informationssuchenden Aufgaben zu unterscheiden wird Text- bzw. Dokumentretrieval oft mit Information Retrieval gleichgesetzt [90]. Textretrieval (TR) Systeme durchsuchen eine Kollektion von Textdokumenten mit dem Ziel, diejenigen Dokumente zurückzuliefern, die einer Abfrage entsprechend als relevant erachtet werden können. Im Gegensatz zu strukturierten und klaren Abfragen, wie etwa auf Datenbanksystemen oder beim DR, beschäftigen sich TR Systeme mit vagen und unvollständigen Abfragen, auf die a priori keine eindeutige Antwort existiert [4].

TR Systeme vergleichen die Abfrage mit allen in der Kollektion vorhandenen Dokumenten. Da diese oft nur teilweise mit einer Abfrage korrelieren, wird ein Wert für die Ähnlichkeit zwischen der Abfrage und einem Dokument errechnet. Die Ergebnisse bezüglich einer Abfrage werden entsprechend ihrem Übereinstimmungsgrad gereiht und an den Benutzer als Resultat zurückgegeben. Dieser trifft anhand einer manuellen Überprüfung eine Auswahl der zurückgegebenen Dokumente. Diese Benutzerauswahl kann, wie zuvor schon angesprochen, zusätzlich als Feedback an das System zurückgegeben werden und für zukünftige Abfragen ähnlicher Art als Lernbeispiel oder Referenz dienen.

Im weiteren Verlauf der Arbeit wird Textretrieval als Synonym für Information Retrieval verwendet.

### 2.3.2 Textfilterung

Im Zuge der Netzwerktechnik erlangte der Begriff der dynamischen Datenverarbeitung immer größere Bedeutung. Eine Vielzahl an Informationen wird täglich an unzählige Systeme übertragen. Durch Entwicklungen wie e-Mails oder Newsgroups im Internet wurden Mechanismen notwendig, mit deren Hilfe neue Informationen ausgewählt und relevantes Material gesammelt werden kann. Bei der Textfilterung (TF), auch bekannt unter den Begriffen Dokumentfilterung, Textrouting oder

Selective Dissemination of Information (SDI) [6, 11], geht es um die Handhabung dynamischer Textressourcen. Man versteht darunter das Aussieben (und Weiterleiten) von Textdokumenten, die aufgrund einer vorangegangenen Analyse für einen Benutzer in Frage kommen [13, 69, 2]. Um entsprechende Benutzerpräferenzen zu beschreiben, werden sogenannte Benutzerprofile angelegt. Ein solches Profil wird mit den beim System ankommenden Dokumenten verglichen. Einem Benutzerprofil entsprechend relevante Dokumente werden vom System erkannt und abgelegt. So könnten beispielsweise Fussball-Zeitungsartikel aus tausenden von täglich übermittelten Artikeln ausgefiltert werden. Die bezogenen Dokumente werden an das entsprechende Benutzerprofil zurückgereicht, wobei diese keiner Reihung unterzogen werden. Die auf diese Art bezogenen Dokumente können ebenfalls direkt an einen Adressaten (System oder Benutzer) weitergeleitet werden, wodurch ein weiterer Aspekt an Dynamik hinzukommt: Informationen kommen nicht nur kontinuierlich beim System an sondern werden auch weitergeleitet.

Diese Aufgabe kann ebenfalls als ein Kategorisierungsproblem gesehen werden [80]. Dokumente werden einer von zwei möglichen Kategorien zugewiesen: entweder zur Kategorie relevanter oder nicht relevanter Dokumente.

Der Unterschied zwischen TF und TR besteht in zweierlei Hinsicht. TR beschäftigt sich mit dynamischen und unspezifischen Abfragen auf einem statischen und unstrukturierten Pool an Informationsquellen. TF hingegen arbeitet mit stabilen und spezifischen Abfragen auf dynamischen und unstrukturierten Informationsressourcen. Generell kann TF als spezieller Fall des TR gesehen werden, wenn der Datenraum beim TR als sehr dynamisch angenommen wird [4]. Abbildung 2.2 illustriert diesen Zusammenhang. Einige Experimente zu diesen Aufgabenbereich finden sich in Oard et. al.[65].

### 2.3.3 Textgruppierung

Die Textgruppierung (oder Dokumentgruppierung) erlangte in den letzten Jahren immer mehr an Bedeutung. Man unterteilt diesen Aufgabenbereich in die eigentliche Dokumentkategorisierung (DK) und das Dokumentclustering (DC). Beide Gebiete finden ihre Anwendung in der Zuweisung von Dokumenten zu bereits vorgegebenen

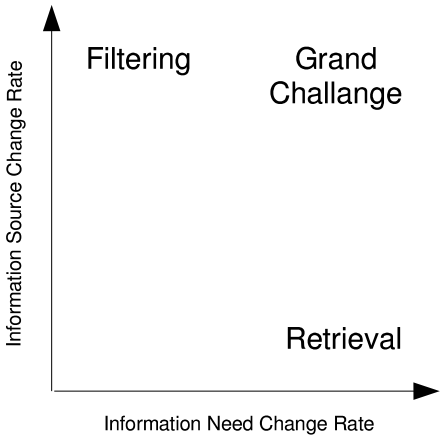



Abbildung 2.2: Textfilterung und Textretrieval, aus [64, Seite 4]

oder automatisch entwickelten Kategorien. Bei der computergestützten DK und dem DC weist, im Gegensatz zum menschlichen Experten, ein Computersystem Dokumenten bestimmte Kategorien zu.

DK zielt darauf ab, Dokumente bestimmten, bereits vordefinierten Kategorien zuzuordnen [27, 51, 17]. Diese Kategorien werden in der Regel von einem Domain-Experten festgelegt und sind somit bereits vor dem eigentlichen Kategorisierungsprozess bekannt. Die einzelnen Dokumente werden während der Kategorisierung einer, mehreren oder keiner dieser Kategorien zugeteilt. Um Dokumentkategorisierungen zu erlernen, werden Systeme entweder mit einer großen Zahl positiver Lernbeispiele [55] oder durch das Feedback eines interaktiven Benutzers trainiert. Um eine entsprechende Kategorisierung richtig zu erlernen, ist jedenfalls eine große Anzahl postitiver Beispiele (Dokumente samt „Labels“) notwendig [17].

DC hingegen beschäftigt sich mit der Gruppierung ähnlicher bzw. verwandter Dokumente in Kategorien, man spricht in diesen Zusammenhang von Clustern [6]. Aus diesem Blickwinkeln betrachtet, arbeitet Dokumentclustering nicht auf einzelnen Dokumenten sondern auf einer Dokumentsammlung, einer Kollektion von Dokumenten (Corpus) [4].

Im Gegensatz zur Dokumentkategorisierung sind die Kategorien (Cluster) nicht im vorhinein bekannt, sondern werden während des Prozesses dynamisch generiert. Die Merkmale der einzelnen Dokumente, anhand derer eine Zuteilung zu einem Cluster durchgeführt wird, werden vom System selbst ermittelt. Es existieren keine positiven

Beispieldaten und jedes Dokument wird dem ähnlichsten Cluster zugewiesen. Es erfolgt also genau eine Zuteilung eines Dokuments zu genau einem Cluster [6].

Generell gesehen können zwei verschiedene Arten des Clusterings [4] verwendet werden. Einerseits kann „globales“ Clustering eingesetzt werden, um Dokumente anhand ihrer Eigenschaften innerhalb der gesamten Kollektion zu gruppieren. Andererseits kann Clustering verwendet werden, um Dokumente anhand einer Abfrage in zwei (echte) Teilmengen aufzuspalten. Eine Teilmenge enthält für diese Abfrage relevante, die andere nicht-relevante Dokumente (siehe Abbildung 2.3).

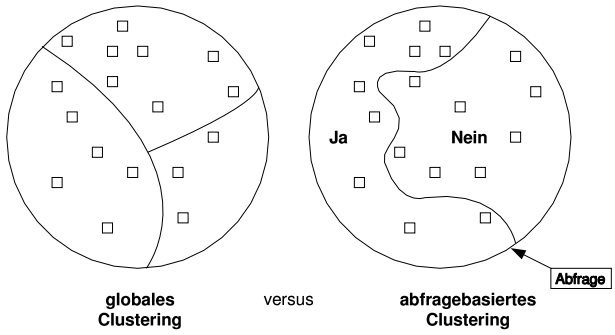



Abbildung 2.3:

Clusteringmethoden werden im IR oft verwendet, um die originale Abfrage auf verschiedene Mengen ähnlicher Dokumente abzubilden [4]. Auf diese Weise können bessere Ergebnisse für den einzelnen Benutzer erzielt werden, der seine Informationsbedürfnisse in Form von vagen Abfragen an das System stellt und ähnliche Dokumente als Ergebnis erwartet. Deshalb kann Clustering auch als Operieren auf gestellten Abfragen verstanden werden.

Sowohl die DK als auch das DC dient der Organisation von Textdokumenten. Ziel ist es einerseits, das Wiederauffinden einzelner Informationen in Bezug auf Geschwindigkeit zu optimieren. Andererseits wird das Ergebnis einer Suche insofern verbessert, als ebenfalls auch andere verwandte Dokumente gefunden werden, die der eigentlichen Abfrage nicht direkt entsprechen.

## 2.4 Repräsentation von Texten

Textdokumente in großen Datensammlungen werden meistens durch eine Menge von Indextermen (auch Schlüsselwörter, Indexwörter, Deskriptoren, ...) repräsentiert.

Diese Indexterme können entweder direkt aus dem Text stammen (automatisch bezogen) oder von einem Domain-Experten vergeben (manuell erstellt) worden sein. Aus welcher Quelle diese Wörter auch stammen, sie stellen eine logische Repräsentation des Textes dar. Man spricht in diesem Zusammenhang von einer deskriptiven Textrepräsentation [4].

Im wesentlichen gibt es zwei Techniken der Indexierung (oder Repräsentation) von Textdokumenten [6, 4]. Einerseits kann eine Menge von inhaltsbildenden Schlüsselwörtern, die in der Regel Domain-Experten erarbeiten, definiert werden. Man spricht in diesem Zusammenhang von einem feststehenden Vokabular (Controlled Vocabulary), anhand dessen der Inhalt von Dokumenten repräsentiert wird. Jedem Dokument der Kollektion werden bestimmte Begriffe dieses Vokabulars zugeteilt, mittels deren der Dokumentinhalt beschrieben wird. Zum anderen kann eine automatische Textindexierung durchgeführt werden. Hierbei wird der gesamte Text des Dokuments als Basis herangezogen, um selbstständig Schlüsselwörter zu finden (Uncontrolled Vocabulary). Indexterme können sowohl explizit als Wörter im Text genannt oder auch, wie etwa durch den Einsatz von Thesauri, implizit abgeleitet werden [6].

Moderne Computer ermöglichen es, Dokumente durch ihren gesamten Wortschatz zu repräsentieren. Man spricht in diesem Zusammenhang von einer Volltext-Repräsentation. Durch den Einsatz von sehr großen Dokumentsammlungen stoßen jedoch auch neuere Geräte schnell an ihre Grenzen. In solchen Fällen muss die Datenmenge während der automatischen Textindexierung reduziert werden. Diese Aufgaben erfüllen Stoppwortlisten (Filterfunktion), Stemmer (Wortnormalisierung) und die Erkennung von Nominalphrasen. Tagger können in diesem Schritt zusätzlich eingesetzt werden, um mittels der Identifikation von syntaktischen Wortkategorien einen Auswahlprozess zu unterstützen. Diese sogenannten Textoperationen ermöglichen eine Reduktion des gesamten Textes auf eine Menge von Indexwörtern (siehe Abbildung 2.4) [4]. Zusätzlich wird der Fokus des Dokumentinhaltes auf die gewählten Indexterme gelegt.

Moderne IR Systeme basieren im Gegensatz zu einzelnen Dokumentanalysen auf einer Corpusanalyse. Die Idee dahinter ist die Extraktion wesentlicher Merkmale einer gesamten Kollektion von Dokumenten, wobei die einzelnen Dokumente und

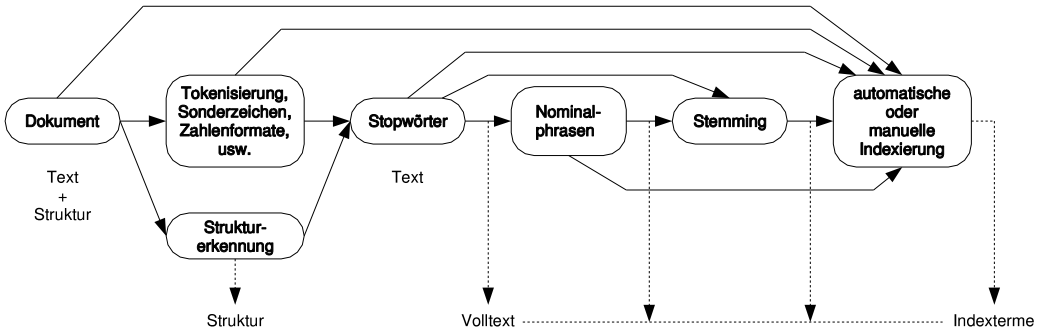



Abbildung 2.4: Textrepräsentation - Vom Volltext zu Indextermen, aus [4, Seite 6]

deren Strukturen ebenfalls nicht vernachlässigt werden.

Wie gerade erläutert werden Texte aufgrund ihres Umfangs und ihrer Anzahl durch eine reduzierte Menge an Indextermen repräsentiert. Diese werden bei der automatischen Indexierung mit heuristischen Methoden, die aus dem Bereich des Natural Language Processings kommen, ermittelt. Anschließend wird für diese Indexterme eine geeignete Repräsentationsform gewählt, um die ermittelten Wörter für das System geeignet darzustellen.

Eine Variante für eine solche Textrepräsentation stellt das Vektorenmodell [73, 6, 4, 74] dar, welches die das Dokument repräsentierenden Indexterme als Vektor dargestellt. Frühere Modelle wie das Boolean Modell [4] verwendeten Binärwerte zur Repräsentation von Indextermen in diesem Vektor. Vorhandene Terme wurden mit 1 gewichtet, alle anderen mit 0. Da eine reine Darstellung der Existenz von Indextermen für die Zweckerfüllung von IR Systemen meist nicht ausreicht, erfolgt zusätzlich eine Giwichtung der Begriffe.

Da sowohl die Termermittlung als auch die Termgewichtung eine zentrale Rolle für die Effektivität von IR Systemen spielen [4], wurden für die Übernahme dieser Aufgaben verschiedene Modelle entwickelt. Im Zuge dieser Arbeit wird insbesondere auf zwei Modelle eingegangen:

Beim Vektorenmodell gibt das Gewicht eines Termes Auskunft darüber, wie oft dieser Term im Dokument vorkommt. Die Ähnlichkeit von Dokumenten untereinander wird anschließend durch die Berechnung des Winkels zwischen zwei Dokumentvektoren angegeben [6, 4, 27, 90].

Einen gänzlich anderen Ansatz verfolgt das Thema-Rhema Modell. Hierbei werden

die Indexterme eines Dokuments nicht als gleichbedeutend behandelt, vielmehr werden Beziehungen zwischen Termen aufgrund der syntaktischen Satzstruktur aufgebaut. Als ein Bereich der Systemic Funktional Grammar [6] ermöglicht dieses Modell eine tiefere Analyse des Textes, wobei das im Text vorhandene Wortmaterial in sogenannte Themata und Rhemata unterteilt wird. In dieser Theorie wird davon ausgegangen, dass das thematische Material neue Informationen des Textes beschreibt, also die Kernaussagen enthält. Dieses thematische Material wird durch das rhematische Material näher bestimmt, erklärt und ergänzt. Es findet im weiteren Sinn eine Konzeptbildung statt [6].

Auf die Arbeitsweise beider Modelle wird im Kapitel 4 genauer eingegangen.

## 2.5 Clusterretrieval

Das Clusterretrieval stellt eine spezielle IR Variante dar. Wie bei anderen IR Verfahren sollen auch hier Dokumente aus einer Dokumentmenge automatisch ausgewählt werden, die für eine gegebene Abfrage relevant sein könnten.

Eine Abfrage wird jedoch nicht mit allen vorhandenen Dokumenten einzeln verglichen. Vielmehr wird von jedem Cluster ein sogenanntes Clusterzentrum oder Cluster-Zentroid berechnet, indem die in einem Cluster befindlichen Dokumente zusammengefasst werden. Die Abfrage wird anschließend nur mit den Clusterzentren verglichen, nicht mehr mit den einzelnen Dokumenten. Als Suchergebnis werden alle Dokumente zurückgegeben, die in relevanten Clustern liegen. Dies entspricht der sogenannten Clusterhypothese [21], die aussagt, dass

„*associated documents tend to be relevant to the same requests*“

In Abbildung 2.5 sind die Dokumente der zu durchsuchenden Dokumentenmenge als Quadrate dargestellt. Nicht alle Dokumente werden hinsichtlich ihrer Relevanz einzeln bewertet, sondern es erfolgt lediglich ein Vergleich der Zentroide der zugehörigen Cluster (= Kreise) mit der Abfrage. Die relevanten Cluster sind grau hinterlegt. Ein Cluster wird als relevant erachtet, wenn er ausreichend ähnlich zur Abfrage ist. Die Zahlenwerte neben den Clustern und unter den Dokumenten geben

den Ähnlichkeitsgrad zur Abfrage an. Hier wurde als Grenzwert für die Ähnlichkeit ein Wert von 0.5 festgelegt, der überschritten werden muss.

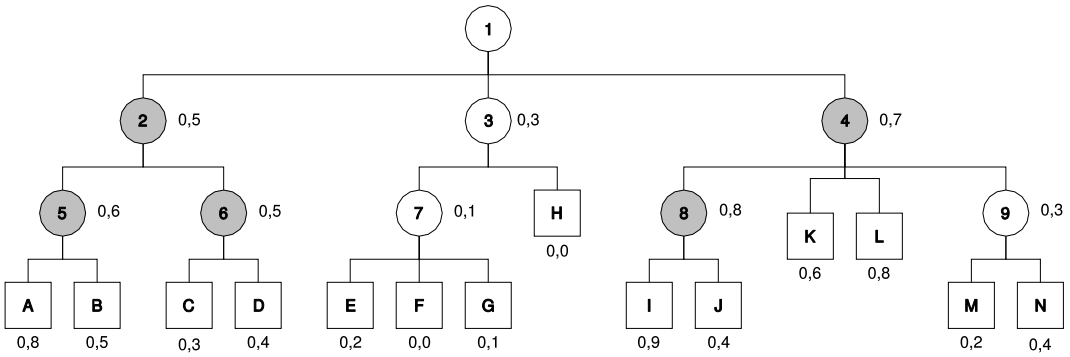



Abbildung 2.5:

In Abbildung 2.5 ist ein hierarchisches Clusteringverfahren angewandt worden, weshalb es Zentroide auf mehreren Ebenen gibt.

Cluster 1 ist der Zentroid für den alles umfassenden Cluster, wie er etwa am Ende eines agglomerativen Verfahrens[1] entsteht. Cluster 2, 3 und 4 sind zwei Schritte vor dem Ende des Clusteringprozesses entstanden und bestehen sowohl aus weiteren Clustern wie auch aus Dokumenten (Mischcluster). Die Cluster 5, 6, 7, 8 und 9 bilden die erste Stufe der Clusterbildung und beinhalten nur Dokumente (reine Cluster).

Um den Suchraum für eine genauere Untersuchung einzelner Dokumente einzugrenzen, wird die gestellte Abfrage mit allen Clustern verglichen. Dokumente aus nicht relevanten Clustern werden bei einer weiteren Untersuchung nicht in Betracht gezogen. In Abbildung 2.5 werden somit die Dokumente E, F, G, H, M und N für eine genauere Untersuchung verworfen. Es kommen nur die Dokumente A, B, C, D, I, J, K und L in Frage.

Die Aufgabe des Clusterings besteht darin, die Anzahl der für das eigentliche Retrieval betrachteten Dokumente zu verringern. Der Vorteil des Verfahrens liegt in der Einschränkung des Suchraumes, da Dokumente irrelevanter Cluster nicht mehr mit der Abfrage verglichen werden müssen.

Außerdem werden bei dieser Art des Retrievals nicht nur die Zusammenhänge zwi-

[1] Agglomerative Clusterverfahren [83, 86] beginnen mit einem Cluster für jedes Dokument. In den Folgeschritten werden Cluster mit ähnlichen Dokumenten in einem übergeordneten Cluster zusammengefasst. Am Ende des Prozesses entsteht ein allesumfassender Cluster.

schen Dokumenten und Abfragen, sondern auch die Zusammenhänge zwischen den einzelnen Dokumenten berücksichtigt, da die Abfrage zum Zeitpunkt des Clusterings noch nicht bekannt war.

KAPITEL 3

# Textgruppierung

Das Aufgabenfeld der Textgruppierung erlangte in den letzten Jahren immer mehr an Bedeutung. In diesem Abschnitt werden die zwei Teilbereiche, die Dokumentkategorisierung und das Dokumentclustering, genauer betrachtet. Um Dokumente miteinander vergleichen zu können, sind sogenannte Ähnlichkeits- oder Abstandsmaße notwendig, die anschließend vorgestellt werden. In Anknüpfung daran erfolgt, zusammen mit diversen Auswertungsmöglichkeiten der erzielten Ergebnisse, die Darstellung verschiedenet Modelle der Dokumentgruppierung. Dadurch wird sowohl eine Bewertung als auch ein Vergleich von verschiedenen Ansätzen möglich.

## 3.1 Dokumentkategorisierung

Unter einer Kategorisierung versteht man eine Abbildung einer Menge $D$ von $n$ zu kategorisierenden Objekten auf eine Menge $K$ von $m$ vordefinierten Kategorien. Diese Abbildung kann anhand einer binären Zugehörigkeitsmatrix, wie sie Tabelle 3.1 zeigt, veranschaulicht werden. Eine 1 im Feld $G^{ij}$ bedeutet, dass das Objekt $i$ in die Kategorie $j$ fällt, eine 0 das Gegenteil.

Die einzelnen Kategorien können als Menge gleichbedeutender Kategorien oder in einer Hierarchie organisiert sein. Ein zu kategorisierendes Objekt kann einer, keiner oder beliebig vielen Kategorien zugewiesen werden [41, 42]. Die Kategorien sind bereits vor dem eigentlichen Kategorisierungsprozess bekannt und von Anwendungsgebiet zu Anwendungsgebiet verschieden, da diese insbesondere auf die Art der Objekte abzustimmen sind.

Ein ideales System für eine Kategorisierung genügt laut Bowker den folgenden Grundsätzen [7]:

Tabelle 3.1: Kategorisierungsmatrix, aus [82, Seite 7], $G^{i,j} \in \{0,1\}$

| | $D^0$ | $D^1$ | $D^2$ | $D^{\cdots}$ | $D^n$ |
|---|---|---|---|---|---|
| $K^0$ | $G^{00}$ | $G^{10}$ | $G^{20}$ | $G^{\cdots 0}$ | $G^{n1}$ |
| $K^1$ | $G^{01}$ | $G^{11}$ | $G^{21}$ | $G^{\cdots 1}$ | $G^{n2}$ |
| $K^2$ | $G^{02}$ | $G^{12}$ | $G^{22}$ | $G^{\cdots 2}$ | $G^{n3}$ |
| $K^{\cdots}$ | $G^{0\cdots}$ | $G^{1\cdots}$ | $G^{2\cdots}$ | $G^{\cdots\cdots}$ | $G^{n\cdots}$ |
| $K^m$ | $G^{0m}$ | $G^{1m}$ | $G^{2m}$ | $G^{\cdots m}$ | $G^{nm}$ |

- **Beständigkeit:** Die gewählten Prinzipien zur Kategorisierung sind beständig, d.h. sie gelten jetzt und für alle Zeit. Dieser Anforderung entspricht z.B. eine chronologische Kategorisierung der eingehenden e-Mails, geordnet nach Datum und Uhrzeit.

- **Eindeutigkeit:** Die gebildeten Klassen schliessen sich gegenseitig aus, d.h. sie sind eindeutig. Es gibt keine Elemente die mehreren Klassen angehören. Als Beispiel kann ein Stammbaum dienen. Es gibt nur eine korrekte Kategorisierung der Mitglieder einer Familie. Jedes Kind hat genau ein leibliches Elternpaar.

- **Vollständigkeit:** Das Kategorisierungssystem ist in sich geschlossen (komplett), d.h. es bietet eine hundertprozentige Abdeckung des Gebietes, das es beschreibt. Diesem Grundsatz genügt z.B. ein alphabetisches Namensverzeichnis aller diplomierten Informatikstudenten im Zeitraum von 1980 bis 2000.

Objekte im Information Retrieval können Texte, einzelne Wörter, Wortgruppen, Bilder, Grafiken, Audiosequenzen, Strukturformeln, uvm. sein. Im weiteren Verlauf dieser Arbeit wird von Textdokumenten als zu kategorisierende Objekte ausgegangen. Die Dokumentgruppierung, wie die automatische Kategorisierung von Textdokumenten genannt wird, untergliedert sich in zwei Teilbereiche, die Dokumentkategorisierung (DK) und das Dokumentclustering (DC).

Die klassische Theorie über Kategorisierungen definiert Kategorien als Container von Objekten mit gemeinsamen Eigenschaften [6]. Ein Objekt kann in Abhängigkeit

seiner Eigenschaften entweder in einem solchen Container sein oder nicht (binär). Diese Theorie schließt jedoch Objekte mit nur teilweisen Übereinstimmungen ihrer Eigenschaften aus. Um diesem Problem entgegenzuwirken, entwickelte man die „Prototypen Theorie“ [48, 66]. Hierbei wird ein Grad der Übereinstimmung von Objekteigenschaften gegenüber einem Repräsentanten einer Kategorie berechnet (fuzzy). Manche Objekte sind aus natürlichen Gründen bessere Repräsentanten einer Kategorie als Andere. Die Zugehörigkeit eines Objekts zu einer solchen Kategorie wird in Bezug auf die Anzahl seiner Gemeinsamkeiten mit dem jeweiligen Repräsentanten ausgedrückt. Diese Definition erlaubt eine polythetische Kategorisierung von Objekten, wodurch Mehrfachzuweisungen über Übereinstimmungsgrade ermöglicht werden ($G^{i,j} \in \{0, 1\}$).

Abbildung 3.1 zeigt den Ablauf des Kategorisierungsprozesses neuer Dokumente. Zuerst werden Dokumente mit existierenden Kategorienzuweisungen an das System gereicht. Während der Trainingsphase generiert das System basierend auf dieser Vorgabe sogenannte Kategorien-Prototypen. In der Testphase wird das System mit unbekannten Dokumenten konfrontiert. Neue Dokument werden den bereits erstellten Kategorien-Prototypen gegenübergestellt und mittels eines Ähnlichkeitsmaßes gewichtet. Anschließend werden dem Dokument jene Kategorien zugewiesen, deren Ähnlichkeit einen bestimmten Grenzwert übersteigt.

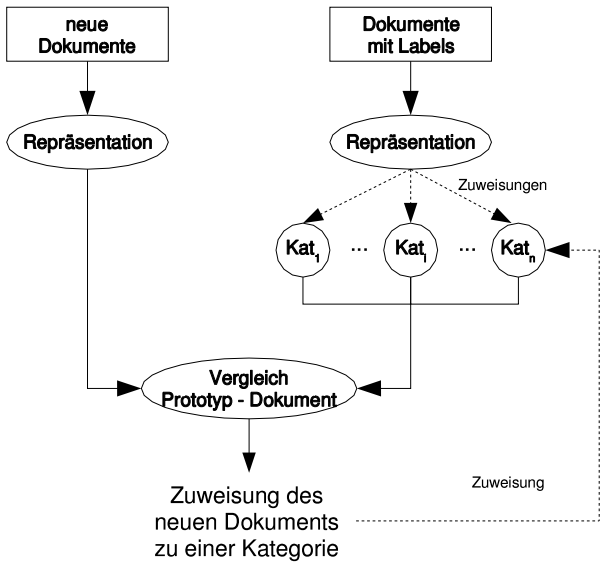



Abbildung 3.1: Dokumentkategorisierung neuer Dokumente

Eine Kategorisierung von Dokumenten kann, wie beispielsweise durch den Ein-

satz Neuronaler Netze, auf zwei Arten erlernt werden: überwacht (supervised) und selbstorganisierend (unsupervised). Beim überwachten Erlernen von Kategorisierungen wird von vordefinierten Kategorien ausgegangen, denen Dokumente zugewiesen werden können. Im Gegensatz dazu werden selbstlernende Systeme als selbstorganisierend bezeichnet. Solchen Systemen liegen keinerlei Informationen über eine mögliche Einteilungsstruktur der Dokumente vor. Sie müssen die Kategorien selbst erzeugen, wobei sie ebenfalls kein manuelles Feedback bekommen. Diese Aufgabe wird generell als Dokumentclustering bezeichnet.

Eine Dokumentkategorisierung bezeichnet eine Zuweisung von Objekten zu Kategorien anhand von Regeln, die entweder vorgegeben oder aus einer Menge von Beispielen entwickelt wurden. Dies kann einerseits durch den Input einer großen Anzahl positiver Lernbeispiele, andererseits auch durch ein Benutzerfeedback zu einzelnen Beispielen erfolgen. Das System erlernt auf diese Weise eine Kategorisierung neuer Dokumente aufgrund beispielhafter Ein- und Vorgaben eines (menschlichen) Lehrers. Im anschließenden Betrieb wird dieses Wissen erinnert und umgesetzt. Im Zusammenhang mit IR spricht man von einer Dokumentkategorisierung.

## 3.2 Dokumentclustering

Als Dokumentclustering bezeichnet man die Vorgehensweise, bei der inhaltsähnliche Dokumente derselben Kategorie zugeordnet werden. Dokumente sollen automatisch und ohne menschliches Einwirken in Gruppen verwandter Dokumente unterteilt werden. Jedes Dokument derselben Kategorie weist dabei ähnliche Merkmale auf und jede Kategorie unterscheidet sich eindeutig von allen anderen Kategorien. Für diese Aufgabe der automatischen Gruppierung von Dokumenten stehen verschiedene Clusteringverfahren zur Verfügung, wie sie in Kapitel 3.4 besprochen werden.

Das Vorgehen beim DC ähnelt einer automatischen Kategorisierung von Dokumenten. Der Unterschied zwischen einer DC und einer DK besteht darin, dass bei der Kategorisierung die Kategorien und ihre typischen Eigenschaften bereits vor der Zuordnung der Dokumente bekannt sind. Zusätzlich stehen bereits Dokumentinformationen zur Verfügung, anhand derer eine richtige Kategorienzuweisung vorge-

nommen werden kann. Beim DC hingegen ergeben sich die Kategorien dynamisch während des Prozesses. Die Verarbeitung läuft gänzlich ohne manuell bereitgestellte Zusatzinformationen ab. Die Eigenschaften der Dokumente, anhand derer ein Clustering vorgenommen wird, sind im vorhinein ebenfalls nicht bekannt. Die Dokumente werden vom System analysiert, wobei diejenigen Dokumenteigenschaften ermittelt werden, die eine ideale Kategorisierung ermöglichen. Das Verfahren muss nicht nur eine Einteilung in Kategorien selbst vornehmen, sondern auch diejenigen Eigenschaften der Dokumente finden, anhand derer möglichst repräsentative Clustereigenschaften abgeleitet werden können.

Die sich daraus ergebende Problemstellung ist die Berechnung der Ähnlichkeit von Dokumenten. Da die Beziehungen zwischen Dokumenten innerhalb einer Kategorie enger sind als zwischen Dokumenten verschiedener Kategorien, ist es notwendig, einen objektiven Vergleichsmechanismus zwischen zwei Dokumenten zur Verfügung zu haben. Dies geschieht durch den Einsatz sogenannter Ähnlichkeits- oder Abstandsmaße (siehe Kapitel 3.3), die auf der Menge der vorhandenen Dokumentrepräsentationen definiert werden.

Da jedes Dokument genau einer Kategorie zugewiesen wird, spricht man nicht von Kategorien sondern von sogenannten Dokument-Clustern. Beim Dokumentclustering werden die Dokumente einer Kollektion in Cluster aufgespalten. Um einen Cluster als eine Menge von ähnlichen Dokumenten adressieren zu können, wird (im Sinne der „Prototypen Theorie“ [6]) für jeden Cluster ein Repräsentant gebildet, der die Eigenschaften aller in diesem Cluster befindlichen Dokumente repräsentiert. Auf diese Weise muss bei einem Vergleich eines neuen Dokuments mit einem Cluster lediglich eine Vergleichsoperation mit dem Repräsentanten (und nicht mit jedem Dokument des Clusters).

Typischerweise sind diese Verfahren sinnvoll, wenn auf der Menge der Dokumenteigenschaften Ähnlichkeitsstrukturen vorhanden sind oder zumindest vermutet werden, die inhaltlich interpretiert werden können. Deshalb eignen sich Clusteringverfahren besonders um große Mengen von Dokumenten zu strukturieren und Beziehungen zwischen den Dokumenten zu erkennen. In Fischer [22] wird ein solches Verfahren auch als „Zusammenzug verwandter Datensätze zum schnelleren Finden“ beschrie-

ben.

Eine wichtige Aufgabe des DC ist die Reduktion des Suchraumes bei Suchoperationen, wie etwa beim zuvor beschriebenen Clusterretrieval (siehe Kapitel 2.5). Durch den Vergleich jedes Dokuments mit einer Abfrage treten bei großen Datenmengen Performanzschwierigkeiten auf. Durch eine Vorabsortierung - einer Gruppierung ähnlicher Dokumente in demselben Cluster - muss eine Abfrage nur mehr mit den jeweiligen Repräsentanten verglichen werden. Sobald ein Cluster mit entsprechenden Eigenschaften gefunden wird, werden alle darin befindlichen Dokumente einer näheren Untersuchung unterzogen.

Das generelle Vorgehen beim Clustering selbst lässt sich in folgende Teilschritte untergliedern:

1. Berechnung der Ähnlichkeit von Dokumenten untereinander: Dieser Schritt ist stark von der Art der Dokumente und deren formaler Repräsentation abhängig. Auf jeden Fall muss ein Ähnlichkeitsmaß (z.B. Skalarprodukt) oder Abstandsmaß zwischen zwei Repräsentationen definiert sein.

2. Erstellung einer Ähnlichkeitsmatrix für alle möglichen Paare aus der Menge der Dokumente (z.B. Dokumentenpaare aus Dokumentensammlung).

3. Berechnung der Cluster auf der Basis dieser Ähnlichkeit: Die Aufgabe besteht darin, $N$ Dokumente in $M$ Cluster aufzuteilen, wobei $M$ entweder bekannt oder unbekannt sein kann. Die Clusterbildung soll so stattfinden, dass die Cluster die zugrundeliegende Struktur der Dokumentmenge repräsentieren.

   Dies wird durch die Forderung erreicht, dass zwei Dokumente eines Clusters in Bezug auf ihr Ähnlichkeitsmaß (Abstandsmaß) näher beieinander liegen müssen als Dokumente aus verschiedenen Clustern.

Die Gestalt der Cluster hängt in erster Linie von drei Einflußgrößen ab. Die Aufbereitung der Daten, etwa die Termgewichtung oder der Einsatz von Stopwortlisten, spielt eine zentrale Rolle bei der Ermittlung der Dokumentrepräsentation (siehe Kapitel 4). Das verwendete Ähnlichkeits- oder Abstandsmaß, welches zwei Dokumentrepräsentationen miteinander vergleicht, hat ebenfalls grosse Auswirkung auf

die Gestalt der Cluster. Und nicht zuletzt ist das verwendete Clusteringverfahren ausschlaggebend für die Bildung der Cluster.

## 3.3 Ähnlichkeits- und Abstandsmaße

Ähnlichkeitsmaße beschreiben die Ähnlichkeit zwischen zwei Objekten (hier Dokumentrepräsentationen) $i$ und $j$ in Bezug auf die untersuchten Dokumenteigenschaften (hier gewichteter Wortvektor).

**Definition 1 – Ähnlichkeitsmaß**

Sei $O$ eine Menge von Objekten und $Q$ das Einheitsintervall $[0, 1]$, dann wird ein Ähnlichkeitsmaß $S$ definiert als:

$$S : O \times O \to Q \text{ , wenn gilt} \tag{3.1}$$

(1) $S(i, j) = S(j, i)$

(2) $S(i, j) = 1 \iff i = j$

(3) $S(i, i) > S(i, j)$ für alle $j \neq i$

Diese Definition eines Ähnlichkeitsmaßes ist abhängig von der gewählten Repräsentationsform der zu vergleichenden Dokumente. Wird beispielsweise eine Repräsentation anhand eines festgelegten Vokabulars verwendet, sind die Bedingungen (1), (2) und (3) nicht unbedingt erfüllt. Zwei nicht identische Dokumente, die dieselben Indexwörter gleich oft im Text haben, würden somit eine Ähnlichkeit von 1 aufweisen und als ident gewertet werden. Bei einer ausreichend genauen Repräsentation sollte dieses Problem nicht auftreten.

Es gibt unzählige Beispiele solcher Ähnlichkeitsmaße, wie unter Anderem der Matching Coefficient, der Jaccard-Koeffizient, das Skalarprodukt, das Kosinusmaß, das Dice-Maß, um nur die Wichtigsten anzuführen [88].

Analog zu einem Ähnlichkeitsmaß ergibt sich eine 'Unähnlichkeit' oder ein Abstand zwischen zwei Dokumenten $A(i,j) = 1 - S(i,j)$. Beispiele für Abstandsmaße beim Clustering sind der Euklidische Abstand und der City-Block-Abstand [88].

Nach der Definition dieser Maße besteht der nächste Schritt des Clusterings in der Erstellung einer Ähnlichkeitsmatrix für alle möglichen Dokumentenpaare der Dokumentenmenge. Für verschiedene Werte einer Ähnlichkeitsmatrix, die aus Ähnlichkeitsmaßen aufgebaut ist, gilt: Je größer das Ähnlichkeitsmaß zwischen zwei Dokumenten ist, desto ähnlicher sind sie. Für verschidene Werte von Abstandsmaßen gilt: Je kleiner das Abstandsmaß zwischen zwei Dokumenten ist, desto ähnlicher sind sie.

Bei der Interpretation solcher Matritzen muss jedoch auf die Spezifika des gewählten Ähnlichkeits- oder Abstandsmaßes Rücksicht genommen werden.

## 3.4 Modelle zur Dokumentgruppierung

### 3.4.1 Statistische Modelle

Statistische Modelle nutzen die Auftrittshäufigkeiten von Termen in Dokumenten, um deren Inhalt zu repräsentieren. Mittels dieser Repräsentationen und den definierten Ähnlichkeits- oder Abstandsmaß werden Ähnlichkeiten zwischen Dokumenten berechnet. Durch die Einführung von Grenzwerten für Übereinstimmungsgrade zwischen Dokumenten werden Texte automatisch gruppiert , wodurch eine DK erfolgt.

Der bekannteste Vertreter ist k Nearest Neighbor (kNN). Er wurde intensiv studiert und oft für Dokumentkategorisierungszwecke eingesetzt. Die Vorgehensweise des Algorithmus ist einfach [94]: Das System findet die $k$ nähesten Nachbarn des Dokuments, das gerade getestet wird und verwendet deren Kategorisierungen, um die Kategorien zu gewichten. Gleiche Nachbarkategorisierungen werden dabei aufsummiert. Am Ende wird eine Reihung der Kategoriengewichte durchgeführt und dem Dokument die Kategorie mit der höchsten Gewichtung zugewiesen. Die Kategorienentscheidung wird dabei über die Formel

$$y(\overrightarrow{x}, c_j) = \sum_{\overrightarrow{d_i} \in kNN} sim(\overrightarrow{x}, \overrightarrow{d_i}) y(\overrightarrow{d_i}, c_j) - b_j \tag{3.2}$$

berechnet [94, 35]. $y(\overrightarrow{d_i}, c_j) \in 0,1$ ist die Kategorisierung des Dokuments $\overrightarrow{d_i}$ in Bezug auf die Kategorie $c_j$ ($y = 1$ für JA, $y = 0$ für NEIN). $sim(\overrightarrow{x}, \overrightarrow{d_i})$ steht für die Ähnlichkeit des Testdokuments $\overrightarrow{x}$ mit dem Trainingsdokument $\overrightarrow{d_i}$. $b_j$ ist ein kategorienspezifischer Grenzwert für die Binärentscheidung.

Weitere Vertreter der statistischen Dokumentkategorisierung sind Support Vector Machines (SVM) [42, 35] und Latent Semantic Indexing (LSI) [81, 82].

### 3.4.2 Probabilistische Modelle

Während statistische Modelle den Dokumentkategorisierungsprozess beschreiben, versuchen probabilistische Modelle dieses Vorgehen zu erklären [4]. Hierbei wird mit Wahrscheinlichkeiten bei der Zuweisung von Dokumenten zu Kategorien gearbeitet. Die Dokumentkategorisierung wird über die Abschätzung der Wahrscheinlichkeit $P(c|d)$ einer Zuweisung eines Dokuments $d$ zu einer Kategorie $c$ berechnet [40, 89]. Die Kategorien eines Dokuments werden anschließend aufgrund des ermittelten Wahrscheinlichkeitswertes in absteigender Reihenfolge sortiert. Je größer der Betrag von $P(c|d)$ ist, desto wahrscheinlicher wird das Dokument $d$ der Kategorie $c$ zugewiesen. Dieses Vorgehen ist auch bekannt unter dem Namen Probability Ranking Principle (PRP) [68].

Bei der Bestimmung der Wahrscheinlichkeitswerte von Dokumentzuweisungen wird diesbezüglich von einer einfachen Berechnung von $P(Kategorie|Dokument)$ ausgegangen. Diese Annahme ist jedoch nicht realistisch, da weder die richtig zuzuweisenden Dokumente noch deren Anzahl a priori bekannt sind [6]. Um dieses Problem zu umgehen wird iterativ vorgegangen. Basierend auf der Auswertung des vorangegangenen Ergebnisses (gestartet wird mit den Lernbeispielen) wird die Kategorienbeschreibung umformuliert, wobei Terme aus zugewiesenen Dokumenten hinzugefügt beziehungsweise Terme aus nicht zugewiesenen Dokumenten entfernt werden.

Ein weiterer Ansatz, der von Goldszmidt [29] beschrieben wurde, beschäftigt sich darüberhinaus mit einer probabilistischen Ähnlichkeitsfunktion für Dokumente, ba-

sierend auf einer Wort-Dokument Abschätzung. Hier wird die Übereinstimmung von zwei Dokumenten $d_i$ und $d_j$ ebenfalls über eine Wahrscheinlichkeitsfunktion berechnet. Dabei dient das Auftreten derselben Wörter im Text als Grundlage.

Eine der bekanntesten Vertreter dieser Modelle ist Naive Bayes [1, 81]. Hier wird das Theorem nach Bayes eingesetzt, um die Wahrscheinlichkeit eines Dokuments zu einer Kategorie abzuschätzen:

$$P(c_j|d) = \frac{P(c_j)P(d|c_j)}{P(d)} \tag{3.3}$$

Der Nenner der obigen Formel unterscheidet nicht zwischen einzelnen Kategorien, weshalb er vernachlässigt werden kann (bleibt konstant). Zusätzlich besteht die Annahme von Wortunabhängigkeiten, weshalb die obige Formel vereinfacht werden kann zu

$$P(c_j|d) = P(c_j)\prod_{i=1}^{M}(t_i|c_j) \tag{3.4}$$

$t_i$ steht in dieser Formel für den das Dokument repräsentierenden Indextermvektor $t$ der Länge $M$. Eine Abschätzung $P'(c_j)$ für $P(c_j)$ kann aus den Trainingsbeispielen, die der Kategorie $c_j$ zugewiesen sind, über

$$P'(C = c_j) = \frac{N_j}{N} \tag{3.5}$$

berechnet werden. Weiters kann eine Abschätzung $P'(t_i|c_j)$ für $P(t_i|c_j)$ durch

$$P'(d_i|c_j) = \frac{1 + N_{ij}}{M + \sum_{k=1}^{M} N_{kj}} \tag{3.6}$$

berechnet werden. $N_{ij}$ stellt die Anzahl des Auftretens des Wortes $i$ in den Dokumenten der Klasse $c_j$ des Traininssets dar.

Ein weiteres Beispiel für ein probabilistisches Modell ist das Bayesian Inference Network (BIN) [11, 12].

### 3.4.3 Genetische Algorithmen

Das Problem des natürlichsprachlichen IR kann letztendlich auf ein Problem der Repräsentation und des Retrievals von Dokumenten überführt werden. Wenn die besten Indexterme gefunden werden, um den Dokumentinhalt zu repräsentieren, reduziert sich diese Aufgabe auf eine Optimierung der Dokumentrepräsentationen. Gerade im Bereich der Optimierung findet das genetische Paradigma seinen Ansatzpunkt.

Genetische Algorithmen stellen neben Genetic Programming, Evolutionary Programming oder Evolution Strategies eine weitere Methode des Evolutionary Computings dar [6]. Sie basieren dabei auf einer bestimmten Anzahl von Schritten [28]:

Eine anfängliche Population von Individuen (Chromosomen) wird entweder zufällig oder heuristisch erstellt. Ein Chromosom ist dabei ein Vektor von Genen, wobei jedes Gen meist durch ein Bit dargestellt wird (es können jedoch auch reelle Zahlen verwendet werden). Die Individuen der aktuellen Population, die Generation genannt wird, werden anhand einer Fitnessfunktion ausgewertet. Die Individuen mit der besten Fitness werden herangezogen, um eine weitere Generation zu bilden. Dabei wird grundsätzlich eine große Anzahl von Individuen selektiert, da mit der Größe einer Gerneration auch die Chance auf „bessere“ Individuen steigt (Ausweitung des Suchraumes).

Zwei Operationen, das Crossover und die Mutation, werden verwendet um neue Individuen zu bilden. Crossover wird auf zwei ausgewählten Individuen, den Eltern, ausgeführt, indem beide Eltern Teile ihrer Chromosomen auf zwei neue Individuen, ihre Kinder, aufteilen. Es findet ein Austausch von Teilketten der Chromosomen nach einem zufällig bestimmten Crossover Punkt statt. Jedes der zwei neu gebildeten Kinder erhält je eine Teilkette der Chromosomen der Mutter und des Vaters. Dadurch bleibt die Gesamtanzahl an Chromosomen konstant und das Erbgut der Eltern wird unverändert an die Kinder weitergegeben. Die Mutation hingegen verhindert ein zu schnelles Konvergieren zu lokalen Maxima (des Suchraumes), indem zufällig ein Gen der Kinder invertiert wird. Dadurch werden neue Individuen geschaffen, die durch reines Crossover nicht entstehen können. Dies geschieht sehr selten mit einer vorgegebenen Wahrscheinlichkeit.

Der Ablauf von genetischen Algorithmen ist in Abbildung 3.2 skizziert.

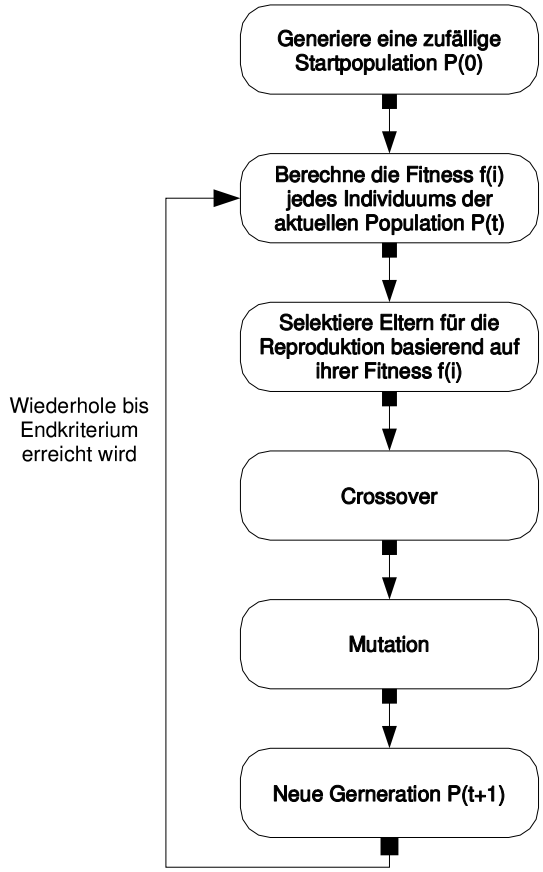



Abbildung 3.2: Arbeitsweise von Genetischen Algorithmen

Genetische Algorithmen stellen einen iterativen Prozess dar, der keine Konvergenz garantiert. Die Abbruchbedingung kann sowohl über die Anzahl der Iterationen als auch über das Erreichen einer bestimmten Fitness erfolgen. Über den Einsatz genetischer Algorithmen für das Problem des Dokumentclusterings sei hier auf weiterführende Litaratur hingewiesen [18, 19, 47, 34].

Um nun genetische Algorithmen für den Einsatz des Clusterings zu adaptieren, muss folgendes festgelegt werden [18]:

1. Eine geeignete Repräsentationsform des Lösungsraumes
2. Eine Fitness-Funktion, die die einzelnen Lösungen bewertet
3. Die genetischen Operationen, die angewendet werden
4. Die Parametereinstellungen (Populationsgröße, Wahrscheinlichkeit für Operationen, ...)

Sobald diese Punkte festgelegt wurden, kann das Clustering beginnen. Seien $N$ Dokumente gegeben, die in $M$ Cluster aufgeteilt werden sollen, so kann eine anfängliche Population zufällig gewählt werden, so dass jeder Cluster im Durchschnitt $\frac{N}{M}$ Dokumente enthält. Der Algorithmus weist anschließend solange Dokumente anderen Clustern zu, bis eine Verbesserung des Clusterings nicht mehr möglich ist. Dies wird mittels der Fitness-Funktion berechnet. Das Ziel ist eine Maximierung der Inter-Cluster Ähnlichkeit, die über die Ähnlichkeit von Dokumenten zueinander berechnet wird.

Jedes Dokument $D$ wird durch einen Termvektor $d_i$ repräsentiert, wobei das $i$-te Element des Vektors das Termgewicht des $i$-ten Indexterms enthält. Auf diesen Dokumentrepräsentationen wird ein Ähnlichkeitsmaß definiert.

Ein Chromosom oder Individuum des genetischen Algorithmus ist eine Dokument-Cluster Zuweisungsreihe [26]. Diese wird als $N$-elementige Kette von Integerwerten repräsentiert, wobei das $i$-te Element der Clusterzuweisung des $i$-ten Dokuments entspricht (siehe Abbildung 3.3).

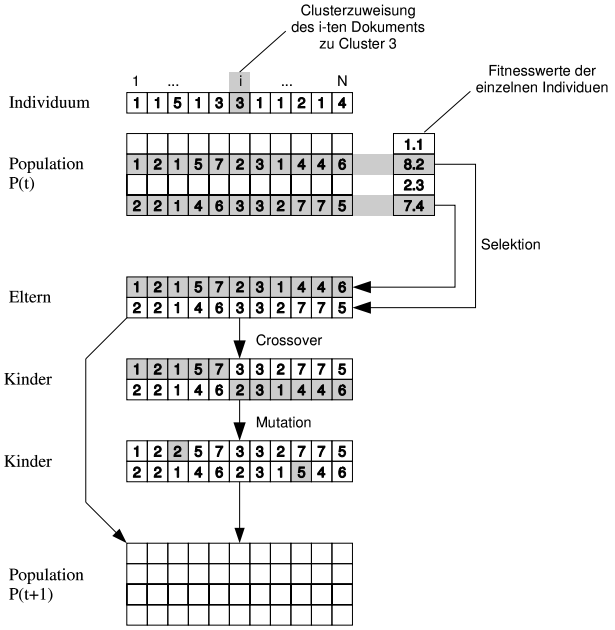



Abbildung 3.3: Dokumentclustering mit genetischen Algorithmen

Die Fitnessfunktion basiert auf der Aufsummierung der Inter-Cluster-Ähnlichkeitsberechnung über eine zuvor definierte Ähnlichkeitsfunktion.

Der genetische Algorithmus erstellt nun (zufällig oder mit heuristischen Methoden)

eine initiale Partition $P(0)$. Für jedes Individuum dieser Partition wird nun die Fitness berechnet. Eine Selektion der Individuen mit der höchsten Fitness führt mit anschliessenden Crossover-Operationen zur Bildung neuer Individuen. Auf diese Individuen wird die Mutation angewandt, um neuartige Individuen zu produzieren, die durch reines Crossover nicht entstehen können. Durch eine anschließende Auswahl wird eine neue Population $P(t+1)$ gebildet. Dieser Prozess wird solange durchgeführt bis ein Abbruchkriterium (maximale Anzahl an Durchläufen, Grenzwert der Fitness-Funktion, ...) erreicht wird (siehe Abbildung 3.2).

### 3.4.4 Clustering Algorithmen

Der weitverbreitetste Repräsentant von Clustering Algorithmen ist $C$-Means bzw. seine erweiterte Variante, Fuzzy $C$-Means [20, 6]. Der $C$-Means Algorithmus zählt zu den selbstlernenden Soft Computing Techniken für die Dokumentkategorisierung.

Der Algorithmus selbst arbeitet folgendermaßen [6]: Zuerst nimmt man eine Unterteilung der zu kategorisierenden Elemente in $C$ Cluster vor. Anschließend werden alle Elemente dem nächstgelegenen Cluster zugewiesen und die Clusterzentren neu berechnet. Diese Schritte wiederholt man, bis sich die Clusterzentren nicht mehr verschieben bzw. unter einem vordefinierten Grenzwert nicht mehr ändern. Der Pseudocode für den Algorithmus sieht wie folgt aus:

1. Wähle eine anfängliche Unterteilung in C Cluster
2. Wiederhole, bis die Clusterzugehörigkeit stabil bleibt

    3. Berechne die Clusterzentren der Unterteilung(en)

    4. Generiere eine neue Unterteilung durch die Zuweisung jedes Elements zum nächstliegenden Clusterzentrum

Ein Problem des $C$-Means Algorithmus ist die im vorhinein definierte Anzahl $C$ an Clustern. Auch eine Abschätzung kann in der Regel nicht im voraus berechnet werden, zumindest ist eine solche nicht trivial.

Eine Erweiterung des C-Means Algorithmus stellt Fuzzy C-Means dar. Hier kann ein Element einem Cluster auch nur teilweise angehören. Dies wird durch einen Wert

zwischen $[0, 1]$ angegeben, der die Zugehörigkeit des Elements zum jeweiligen Cluster ausdrückt. Eine 1 bedeutet, dass ein Element einem Cluster vollständig zugeordnet wird. Eine 0 stellt dar, dass ein Element einem Cluster nicht zugeordent wird. Jeder Wert dazwischen gibt eine teilweise Übereinstimmung mit den Kriterien des Clusters an.

### 3.4.5 Neuronale Netze

Künstliche Neuronale Netze (NN) stellen einen der populärsten Ansätze im Bereich des automatisierten Lernens dar. Sie orientieren sich dabei an ihrem Vorbild, dem menschlichen Gehirn. Es besteht aus Billionen von Neuronen, die wiederum miteinander über Synapsen in Verbindung stehen. Jedes Neuron kann als eine einzelne Verarbeitungseinheit gesehen werden, die auf ankommende Reize bestimmte Outputsignale als Reaktion sendet. Dieses ausgehende Signal dient wiederum anderen Neuronen als Input. Dieser Prozess kann sich über mehrere Schichten von Neuronen ausbreiten, der auch unter dem Namen „Spread Activation Process“ bekannt ist [6]. Auf diese Weise kann das Gehirn wahrgenommene Reize der Sinnesorgane als Informationen verarbeiten und entsprechend darauf reagieren.

Eine künstliches Neuronales Netz ist eine sehr vereinfachte, graphische Repräsentation der unzähligen Neuronen und deren Verbindungen untereinander unseres Gehirnes. Die Knoten solcher Graphen entsprechen den Verarbeitungseinheiten, die Kanten den synaptischen Verbindungen zwischen ihnen. Um die zeitlichen Änderungen der Stärke dieser synaptischen Verbindungen darzustellen, werden die Kanten im Graphen gewichtet. Der Zustand eines Knotens wird durch sein Aktivierungsniveau bestimmt, das eine Funktion des aktuellen Zustands und des vorhandenen Inputs ist. In Abhängigkeit des Aktivierungsniveaus sendet der Knoten ein weiteres Signal an alle seine Nachfolger. Die Stärke des ankommenden Signals hängt dabei von der Gewichtung der Kante ab, die das Signal überträgt.

Die Mächtigkeit künstlicher Neuronaler Netze liegt in ihrer Fähigkeit zu Lernen. Dabei stehen verschiedene Algorithmen zur Verfügung, die in drei Hauptklassen untergliedert werden können: überwachtes (supervised), selbstorganisierendes (unsupervised) und bestärkendes (reinforced) Lernen [6]. Das Lernen selbst erfolgt durch eine

Anpassung der Gewichte an den Kanten. Dies geschieht während der Lernphase, der ersten Phase im Lebenszyklus eines NNs. Die zweite Phase stellt die Anwendung des NNs dar, bei der das erlernte Wissen umgesetzt wird. In der Regel wird ein NN nach Abschluss der Trainingsphase, wie andere auf maschinellen Lernen basierende Systeme, zu einer Black Box, die ein nachträgliches Ändern der Kantengewichtungen schwierig macht.

Es gibt eine Vielzahl verschiedener Neuronaler Netztypen. Die bekantesten sind Perzeptrone, Hopfield Netze, Self-Organizing Maps (SOM), Adaptive Resonance Theory (ART) Netze und Recurrent Netze. Unterschieden werden können sie durch die Art des Lernens (supervised, unsupervised, reinforced), die Art der Verbindungen (Feedback Verbindungen), die Architektur (eine Schicht, mehrere Schichten) und die Art des Inputs (binär, kontinuierlich, bivalent). Einen guten Überblick findet man in [43, 56, 37, 70].

NN sind bekannt dafür, gute Mustererkenner zu sein,daher liegt es nahe, sie auch im Bereich der Textgruppierung einzusetzen. Darüber hinaus simuliert das Neuronale Netzwerk Paradigma sowohl das Vektorenmodell (durch die Verwendung eines Ähnlichkeitsmaßes zwischen einem Inputvektor und gespeichterten Gewichtsvektoren) als auch das probabilistische Modell (durch die Berechnung der Relevanz von Repräsentationen) [6].

In Systemen, die auf dieser Technologie basieren, stellen die Neuronen sowohl die Kategorien (Output-Schicht) als auch die Informationsbeschreibungen (Dokumentterme, Input-Schicht) dar. Eine Kategorisierung erfolgt durch die Aktivierung von Kategorien durch die im Dokument enthaltenen Terme. Die Aktivierung wird dabei von den Neuronen der Inputschicht über die Verbindungen auf die Neuronen der Vergleichs-Schicht weiter an die Neuronen der Kategorien-Schicht propagiert (siehe Abbildung 3.4). Bevor das Netz eine Kategorisierung durchführen kann, werden die gewichteten Kanten, die die eigentliche Information über die Kategorisierung tragen, mit einer Anzahl von Beispieldaten berechnet. Am Ende des Trainings kann das Netz im Testmodus gestartet werden, um das gespeicherte Wissen abzurufen.

In der Regel werden drei Schichten verwendet: eine für die Dokumentterme, eine Zwischenschicht für die Speicherung der Kategorieninformationen und eine Dritte für

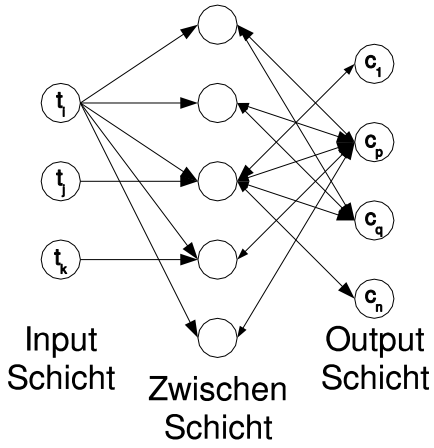



Abbildung 3.4: Neuronales Netz zur Dokumentkategorisierung

die Kategorien selbst. Die Schicht der Dokumentterme dient hierbei als Inputschicht. Es werden die im aktuellen Dokument auftretenden Termgewichte angelegt und der Inferenzprozess gestartet. Danach feuern die Indexterme und generieren Signale zur Kategorien-Schicht. Hiermit endet die Arbeitsphase des NN. Wie in Abbildung 3.4 gezeigt, entwickeln sich die Signale vom Input zum Output, also von links nach rechts.

Das erste und einfachste NN, welches für den Bereich der Dokumentkategorisierung eingesetzt wurde, war das (Multi-Layer-)Perzeptron. Diese Netze basierten auf überwachtem Lernen anhand von Beispielen oder einem Benutzerfeedback, wobei die Muster zu bestehenden Kategorien gespeichert und erinnert wurden. Das Netz konnte keine Kategorien selbst entwickeln, weshalb diese Methode auf eine reine Dokumentkategorisierung eingeschränkt war.

Erst später wurden selbstlernende oder selbstorganisierende Netze entwickelt, die eigenständig Kategorien generieren konnten. Dadurch wurde das Aufgabengebiet des Clusterings für NN zugänglich. Solche Netze, wie sie in dieser Arbeit zum Einsatz kommen, scheinen eine angebrachte Möglichkeit darzustellen, die Probleme verschiedener anderer Clusteringverfahren zu umgehen. Insbesondere sind zwei Arten NN von besonderer Bedeutung für diese Problemstellung:

- **Self Organizing Maps (SOM)** von Kohonen [45, 56] werden weitverbreitet und lange eingesetzt, um ein Dokumentclustering durchzuführen. Eine Kohonen-Karte ist eine zweidimensionale Darstellung eines n-dimensionalen

Termvektors (einer Dokumentrepräsentation). Die Vorteile liegen in der Effizienz und der Darstellungskraft der Ergebnisse. Dokumente werden durch Termvektoren repräsentiert und dem SOM als Input übergeben. SOM berechnet den Gewinner unter den Kategorien und erhöht sein Gewicht als auch die Gewichte der Nachbarn des Gewinners. Dieser Prozess wird wiederholt bis das Netz konvergiert, d.h. die Änderungen kleiner als ein vordefinierter Grenzwert sind. Ähnliche Dokumente häufen sich im gleichen Gebiet dieser Karte. Dokumente (und ähnliche Dokumente) zu suchen, entspricht einer Auswahl eines Gebietes auf dieser Karte. Dennoch generieren diese Netze Cluster, die nach Beendigung des Trainings nicht weiter entwickelt werden können, wodurch ein gesamtes Neutraining unumgänglich wird.

- **Adaptive Resonance Theory (ART)** [56] Netze stellen einen neuen Ansatz für den Einsatz Neuronaler Netze zum Zweck des Textclusterings dar. Anderen Typen Neuronaler Netze ist es nicht möglich, neue Muster zu erlernen, sobald diese sich im Einsatz befinden (beispielsweise bei SOM). ART Netze sind diesbezüglich die Einzigen, die im nachhinein (und auch während des Betriebs) neue Muster erlernen können. Es ist für den Benutzer jederzeit möglich, in den Lernmodus umzuschalten, um das System weiter zu trainieren.

Das Thema der Wartung dieser NN (hinzufügen neuer Dokumente, hinzufügen neuer Terme) wird allerdings nur spärlich in der Literatur behandelt. Die Stabilität NN bei großen Kollektionen von Dokumenten ist ebenfalls schwer zu handhaben [6]. Die Organisation der Cluster ist abhängig von der Reihenfolge der Inputdaten. Cluster können ebenfalls sehr groß werden, wodurch die Karte unbalanciert wird. Weiters ist das Problem der lokalen Minima, wie sie beim Einsatz von SOM auftreten können, zu beachten. Eine Lösung dieses Problems wurde durch Adaptive Resonance Theory (ART) Netze erzielt. SOM ist gut geeignet, um zweidimensionale Karten zu entwickeln. Um höhere Dimensionen einzusetzen, sind jedoch große Anpassungen sowohl der Netze als auch der Darstellung von Nöten.

Es gibt keinen Beweis dafür, dass sich NN im Bereich des IR besser erweisen als alternative Modelle, jedoch stellen sie eine weitere interessante Möglichkeit dar. Auf diese Weise werden auch Dokumente berücksichtigt, die im vorhinein keine direk-

ten Eigenschaften mit den zu vergleichenden Dokument gemeinsam hatten. Diese Vorgehen ist besonders im Aufgabenfeld des IR wünschenswert [6].

## 3.5 Evaluation von Dokumentgruppierungen

Aus der Notwendigkeit heraus verschiedene Modelle der Dokumentgruppierung miteinander zu vergleichen, wurden Metriken und Kennzahlen entwickelt, um die Ergebnisse der einzelnen Modelle in Zahlen auszudrücken und diese somit vergleichbar zu machen.

Aufgabe dieser Metriken ist es, die Effektivität der verwendeten Techniken feststellen zu können. Da sowohl die Kategorien nicht exakt gefasst werden als auch die Dokumente selbst keine exakten Zuweisungen zu Kategorien erlauben, wird eine Rangliste erstellt. Diese gibt an, inwieweit Dokumente mit einzelnen Kategorien übereinstimmen. Eine solche Reihung spielt im Bereich der Textgruppierung eine zentrale Rolle und muss bewertet werden können.

Zwei der bekanntesten Kennzahlen dafür sind Recall und Precision [6, 81, 51, 83, 1, 35, 94, 82, 55, 17].

$$\text{Precision} = \frac{\text{Anzahl der bezogenen, relevanten Dokumente}}{\text{Anzahl der bezogenen Dokumente}} \tag{3.7}$$

$$\text{Recall} = \frac{\text{Anzahl der bezogenen, nicht relevanten Dokumente}}{\text{Anzahl der nicht relevanten Dokumente}} \tag{3.8}$$

Diese Kennzahlen sind weit verbreitet und werden fast immer verwendet, um Textgruppierungen zu bewerten. Der optimale Wert für Recall und Precision ist 1. Wie aus den Formeln ersichtlich, verläuft die die Entwicklung dieser beiden Kennzahlen jedoch invers, weshalb meist ein Kompromiss zwischen ihnen eingegangen werden muss.

Neben Recall und Precision gibt es weitere Kennzahlen, die zur Evaluation herangezogen werden können, wie z.B. Accurrancy, Fallout und Error. Ebenfalls sind Kennzahlen vorhanden, die eine Kombination aus den bereits angesprochenen Metriken Precision und Recall errechnen, wie beispielsweise das harmonische F-Measure oder

das E-Measure [6, 88, 81, 71, 38, 54, 16, 96, 17, 93, 60, 1]. Auch die Breakeven Point Analyse, dem Punkt an dem Recall und Precision den gleichen Wert annehmen, ist von Bedeutung.

Zusätzlich gibt es zwei unterschiedliche Methoden zur Berechnung dieser Kennzahlen [82, 6, 51, 27]. Micro-Averaging berechnet dabei für jede einzelne Kategorie oder für jeden Cluster eigene Werte. Am Ende der Prozedur wird der Mittelwert der einzelnen Ergebnisse gebildet, um das Gesamtergebnis zu erhalten. Beim Macro-Averaging hingegen wird für die gesamte Kollektion gemeinsam die entsprechende Kennzahl berechnet.

Textgruppierungen werden durch den Einsatz von Corpora beurteilt. Ein Corpus besteht aus Dokumenten zusammen mit deren (menschlicher) Kategorisierung. Das Corpus wird zufällig in zwei Teile geteilt, z.B. ein Teil umfaßt 70%, der andere 30% der Dokumente. Das System wird anschließend mit dem größeren Teil trainiert und erlernt eine Kategorisierung. Danach wird der kleinere Teil verwendet, um die erlernte Kategorisierung zu überprüfen.

Eine spezielle Variante zur Beurteilung von Textgruppierungen mittels der oben genannten Metriken stellt die sogenannte $k$-fold Cross Validierung [5, 20] dar. Hierbei wird das gesamte Corpus in $k$ gleiche Teile geteilt. Das System wird anschließend mit $k-1$ Teilen trainiert und mit dem übriggebliebenen Teil bewertet, sodass schließlich mit jedem der $k$ Teile einmal überprüft wurde. Am Ende wird der Durchschnitt des Ergebnisses gebildet.

Mehr zum Thema der Evaluation findet sich im Kapitel 6.

KAPITEL 4

# Dokument-repräsentation

Eine der bedeutensten Einflußfaktoren für IR Systeme ist die Wahl der Dokumentrepräsentation, da alle Operationen auf diesen Repräsentationen durchgeführt werden. Dieses Kapitel behandelt die Möglichkeiten und das generelle Vorgehen bei der Bildung von solchen Repräsentationen. Fuzzy-Sets (siehe Anhang A) können bei der Bildung solcher Repräsentationen eingesetzt werden, um unscharfe Beziehungen und teilweise Übereinstimmungen von Konzepten zu verarbeiten. Im Anschluß daran wird auf die Methoden der Textanalyse im Detail eingegangen. Um das Ergebnis dieser Textanalyse für den Einsatz am Computer aufzubereiten sowie die Bedeutung einzelner Begriffe/Konzepte zu betonen, werden Gewichtungen vergeben. Im Zuge dieser Arbeit kommen zwei verschiedene Gewichtungsverfahren zum Einsatz: Einerseits werden die Gewichte anhand eines Standardmodells, des Vektorenmodells, berechnet. Andererseits wird eine neue Gewichtungsstrategie, basierend auf der Thema-Rhema Theorie, verwendet.

## 4.1 Überblick

Ein wichtiger Punkt für den Erfolg von Information Retrieval Systemen stellt die Repräsentation der Dokumente dar [6]. Die Identifikation und Abbildung der zugrundeliegenden Informationen in Dokumenten spielt hierbei die zentrale Rolle. Das Analyseverfahren muss sorgfältig und vor dem eigentlichen Clusteringprozess festgelegt werden, da es für das Ergebnis ausschlaggebend ist.

Im Bereich der natürlichen Sprache ergeben sich jedoch verschiedene Schwierigkei-

ten beim Auffinden von bedeutenden Informationen innerhalb von Dokumenten. Einerseits erlaubt das grammatikalische Regelsystem unterschiedlichste Positionierungen von Elementen innherhalb von Sätzen. Andererseits ist die Ambiguität (Mehrdeutigkeit) der Sprache ein weiterer nicht zu vernachlässigender Punkt, auf den während der Analyse von Textdokumenten achtgegeben werden muss. Satzelemente oder Phrasen, die wiederum mittels grammatikalischer Regeln aus Wörtern zusammengesetzt sind, können in verschiedenen Kontexten unterschiedliche Bedeutungen zum Ausdruck bringen. Selbst einzelne Wörter tragen oft mehrere unterschiedliche Bedeutungen, die mittels Verfahren wie Word Sense Disambiguation (WSD) oder Latent Semantic Indexing (LSI) behandelt werden [82, 81].

Eine weitere Schwierigkeit stellt die Dimension des Vokabulars dar. Durch die große Anzahl verschiedener Terme einer gesamten Kollektion sind die meisten auf maschinellen Lernen basierenden Systeme überfordert. Ein Problem, das sich somit ergeben hat, ist unter dem Namen „Curse of Dimensionality“ oder „Overfitting Problem“ [82, 80, 55, 4, 6, 81] bekannt. Da einzelne Dokumentrepräsentationen nur einen sehr geringen Anteil des Gesamtvokabulars (Feature-Vektor) enthalten, besteht die Möglichkeit, dass eine Dokumentrepräsentation zu speziell wird und somit keiner anderen Dokumentrepräsentation ähnelt. Dies hat natürlich gravierende Auswirkungen beim Dokumentclustering, das auf der Ähnlichkeitsbestimmung von Dokumenten untereinander aufbaut.

Aus diesem Grund wurden Mechanismen geschaffen, die sich mit der Reduktion des vorhandenen Vokabulars auf eine möglichst kleine und repräsentative Menge beschäftigen. Ziel ist es, durch das Negieren von (möglichst unbedeutenden) Termen die Größe des Vokabulars zu verringen, wobei der Informationsverlust so gering wie möglich gehalten werden soll. Methoden dazu werden in Kapitel 4.2 erläutert.

Bei der Reduktion der Dimensionalität können prinzipiell zwei Standpunkte vertreten werden [14, 60]:

- Bei der lokalen Dimensionsreduktion (LDR) wird für jede Kategorie eine bestimmte Anzahl von relevanten Termen bestimmt, die eine Kategorie definieren. Jedes Dokument der Kollektion hat verschiedene Repräsentationen in verschiedenen Kategorien, d.h. jedes Dokument hat soviele Repräsentationen

wie Kategorien vorhanden sind. LDR wird meist eingesetzt, wenn in der Kollektion von lokalen Korrelationen in der Datenmenge ausgegangen wird.

- Die globale Dimensionsreduktion (GDR) hingegen wählt eine Menge von Termen aus der Gesamtheit des Corpus aus. Jedes Dokument hat genau eine Dokumentrepräsentation, die in jeder Kategorie gleichermaßen verwendet wird. GDR wird generell verwendet, wenn die Ausgansdaten global korrelieren.

Die Reduktion des Feature-Vektors kann auf verschiedene Weise erfolgen [6, 67, 82, 81, 49, 53]:

- Feature Reduction selektiert relevantere Indexterme aus den Vorhandenen, indem weniger unterscheidende Terme zur Bildung des Indexes nicht miteinbezogen werden. Dies geschieht durch den Einsatz von Stopwortlisten, in denen die auszufilternden Terme festgehalten werden. Meist handelt es sich hierbei um Funktionswörter wie Artikel oder Präpositionen, denen keine eigenständige Bedeutung zugeschrieben wird.

- Feature Extraction extrahiert bedeutende Indexterme aus den Vorhandenen. Das Ergebnis dieses Prozesses stellt eine Menge neuer Terme dar, die nicht explizit im Text vorhanden sein müssen (aber können). Man kann dieses Vorgehen auch als eine Art Konzeptbildung ansehen, da verschiedene im Text vorkommende Begriffe auf Abstraktionen zurückgeführt werden können. Beispielsweise kann „Auto“ und „Motorrad“ auf „Verkehrsmittel“ umgelegt werden.

Bei der automatischen Indexierung benutzt man die im Text selbst enthaltenen Wörter zur Bildung der Repräsentation. Nicht alle Wörter tragen jedoch den gleichen Informationsgehalt. Wie auch sonst im IR üblich, sollten bestimmte inhaltsneutrale Wörter (Funktionswörter, Artikel, u.s.w.) aus dem Text ausgefiltert werden [4].

Diese Stopwörter können beim Clustering das Ergebnis deutlich verfälschen, da sie häufig und in sehr vielen Texten vorkommen und wichtigere Clusterstrukturen überdecken. Die Wahl der Einträge der Stopwortliste hängt wiederum von der Art und dem Inhalt der betrachteten Dokumente ab. Im gewissen Rahmen stellt die Auswahl

der Stopwörter bereits eine Vorklassifikation der Dokumente anhand der erwarteten Kategorien dar.

Ferner treten Wörter nicht unabhängig voneinander auf. In der Regel sind bestimmte Wortgruppen häufiger gemeinsam anzutreffen als andere. Diese Eigenschaft kann ausgenutzt werden, um die Anzahl der Dimensionen des zu untersuchenden Vektorraums zu verringern, indem mittels mathematischer Methoden diese Abhängigkeiten ermittelt und in Term-Dokument-Matrizen entsprechend umgerechnet werden. Diese Technik wird oft eingesetzt, um Thesauri automatisch zu generieren. Ebenfalls kann auch der Wortnormalisierung, dem Stemming, dieselbe Rolle beim Dokumentclustering zukommen [4].

Der Einsatz von Mechanismen zur Reduktion des Vokabulars schließt jedoch das Aussieben von relevanten Informationen nicht aus. Auch geht durch diese Filterung von Wortgruppen wie Konjunktionen oder Präpositionen semantisch bedeutendes Material verloren. Man denke nur beispielsweise an die Konstruktionen „das Geschenk VON Hannes“ und „das Geschenk FÜR Hannes“.

Durch die Repräsentation anhand eines eindeutigen Vokabulars kann ein hoher Grad an Exaktheit erreicht werden. Damit ergeben sich jedoch neue Schwierigkeiten bei der Indexierung, da verschiedene Schreibweisen der einzelnen Wörter möglich sind. Dieses Problem wird meist durch den Einsatz von Thesauri bewältigt, die Beziehungen zwischen Wörtern (z.B. Synonym, Homonym, ...) enthalten. Dadurch werden Rückführungen verschiedener Synonyme auf ein und dasselbe Konzept möglich.

Ebenfalls bedacht werden müssen die grammatikalischen Ausprägungen von Wörtern. So meinen beispielsweise die Begriffe „Mannes“, „Mann“ und „Männer“ dasselbe. Durch den Einsatz von Stemmern, die eine Wortnormalisierung durchführen, können alle Begriffe auf eine gemeinsame Stammform zurückgeführt werden, wodurch das Vokabular beträchtlich verringert werden kann.

Um die Auftrittshäufigkeiten von Wörtern innerhalb eines Textes zu berechnen, kann Zipf's law [98, 30] als näherungsweises Modell herangezogen werden, um die Verteilung der verschiedenen Wörter zu bestimmen. Die Regel sagt aus, dass das $i$-häufigste Wort $\frac{1}{i^\theta}$ mal der Frequenz des am häufigsten auftretenden Wortes ist. Das impliziert, dass in einem Text von $n$ Wörtern mit einem Vokabular von $V$

verschiedenen Wörtern das $i$-häufigste Wort $\frac{n}{i^\theta H_V(\theta)}$ mal auftritt, wobei $H_V(\theta)$ die harmonische Reihe der Ordnung $\theta$ von $V$, definiert als

$$H_V(\theta) = \sum_{j=1}^{V} \frac{1}{j^\theta} \tag{4.1}$$

ist, so dass die Summe aller Frequenzen $n$ ist. Im Durchschnitt liegt der Wert von $\theta$ zwischen 1.5 und 2.0. In Abbildung 4.1 sieht man eine typische Verteilung der nach Frequenzen sortierten Wörter.

Ein weiterer Punkt ist die Anzahl der verschiedenen Wörter in einem Dokument, das Vokabular. Zur Vorhersage der Größe des Vokabulars in einem natürlichsprachlichen Text kann das sogenannte Heap's law [36] verwendet werden. Diese Regel, die recht genaue Abschätzungen liefert, besagt, dass das Vokabular $V$ eines Textes von $n$ Wörtern $V = Kn^\beta = O(n^\beta)$ ist, wobei $K$ und $\beta$ entsprechend auf der Textsorte beruhen. Die rechte Seite der Abbildung 4.1 veranschaulicht die Entwicklung der Vokabulargröße in Bezug auf die Größe des Textes. Gängige Werte für $K$ befinden sich zwischen 10 und 100, $\beta$ nimmt einen positiven Wert kleiner 1 an.

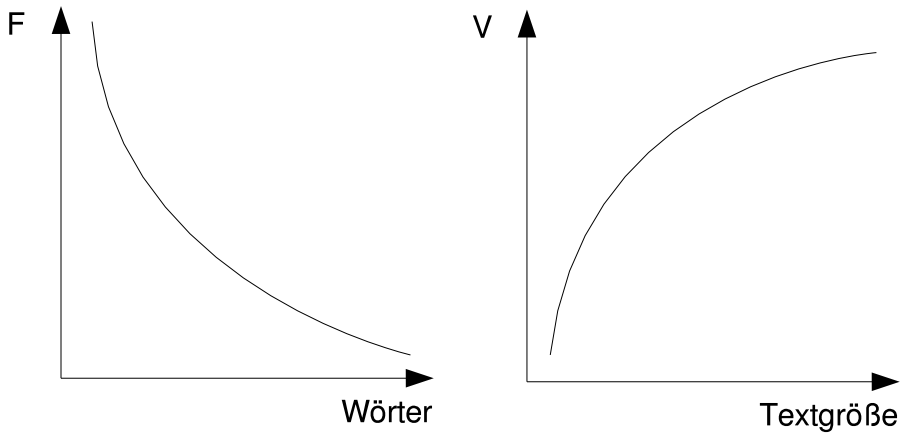



Abbildung 4.1: Verteilung sortierter Wortfrequenzen (links) und Größe des Vokabulars (rechts), aus [4, Seite 147]

Im Folgenden werden einzelne Methoden zur Reduktion des Vokabulars erläutert.

## 4.2 Methoden der Textanalyse

### 4.2.1 Grundlagen

Für die semantische Repräsentation des Inhalts eines Dokuments sind nicht alle Wörter gleich bedeutend. In der geschriebenen (und auch der gesprochenen) Sprache transportieren einige Wortfamilien mehr Bedeutung als andere. Meistens sind es Nomen (oder Nominalgruppen), die den Dokumentinhalt am Besten charakterisieren.

Aus diesen Gründen werden die Textdokumente zuerst aufbereitet, um die Indexterme zu identifizieren. Während des Prozesses können auch andere sinnvolle Textoperationen wie Stopworteliminierung, Wortformenrückführung (Stemming) oder Thesauri (zur Auflösung von Synonymen) verwendet werden, auf die im weiteren Verlauf dieses Kapitels eingegangen wird.

Eine Dokumentrepräsentation anhand von Indextermen führt jedoch zu einer eher ungenauen Repräsentation der Semantik von Dokumenten innerhalb einer Kollektion [4]. Beispielsweise kommt dem Term „Computer“ in einem informatischen Kontext keine besondere Bedeutung zu, aber er führt zu einem Retrieval von Dokumenten, welche mit einer gestellten Abfrage überhaupt nicht übereinstimmen. Ein Inbetrachtziehen von allen Wörtern der gesamten Kollektion als Repräsentation bedeutet also ein zu großes „Rauschen“ für die Aufgabe des IR. Ein Weg dies zu verhindern, ist die Reduktion der Menge der Wörter, anhand derer der Index gebildet wird. Aus diesem Blickwinkel kann eine Aufbereitung als ein Prozess zur Regulierung des verwendeten Vokabulars (Anzahl der verschiedenen eindeutigen Terme des Index) gesehen werden, wodurch eine bessere Performanz des Retrieval Systems erreicht werden kann.

Da eine Kontrolle des Vokabulars eine gängige Technik ist, muss dennoch eine Hand in Hand gehende Eigenschaft im Indexprozess mit einbedacht werden, der vom Benutzer oft nur schwer wahrgenommen wird. Als Ergebnis kann ein Benutzer dadurch überrascht werden, dass einzelne erwartete Dokumente nicht bzw. andere, nicht erwartete Dokumente, sehr wohl bezogen wurden. Möglicherweise erinnert er sich, dass ein Dokument den Text „das Haus des Herren“ beinhaltet, aber dieses

Dokument nicht unter den Top 20 der zurückgelieferten Dokumente ist (da weder „das“ noch „des“ zum Index beitragen). Daher wird klar darauf hingewiesen, das eine Aufbereitung neben der Leistungssteigerung des IR Systems ebenso auch Interpretationsschwierigkeiten des Ergebnisses für den Benutzer bedeuten. Aus diesem Grund haben einige Internet-Suchmaschinen den Einsatz von Textoperationen aufgegeben, da es dem gewöhnlichen Benutzer nur schwer möglich ist, das Ergebnis richtig zu interpretieren [4]. Die Idee dahinter ist ein einfacherers IR, das zwar auf einem verrauschten Index basiert, dafür aber intuitiv vom Benutzer gedeutet werden kann, da es als Volltextsuche interpretiert wird.

Zur Verbesserung des Ergebnisses können neben der Aufbereitung von Dokumenten können auch andere Operationen auf den Texten ausgeführt werden. Hierunter fällt beispielsweise die Auflösung der Synonymität von Wörtern durch den Einsatz von Thesauri.

Die Aufbereitung der Daten kann, ähnlich wie auch in [4] beschrieben, in vier verschiedene Textoperationen (oder Texttransformationen) unterteilt werden:

1. Lexikalische Analyse des Textes mit dem Hauptaugenmerk auf der Behandlung von Zahlen, Bindestrichen, Satzzeichen und Buchstaben (Sonderzeichen).

2. Bestimmung der Wortkategorien der im Text vorkommenden Wörter durch ein Tagging, um semantisch wertvolle Indexterme zu identifizieren.

3. Eliminierung von Stopworten mit dem Hauptaugenmerk auf ein Ausfiltern von Wörtern, die keinen Einfluß auf das Clustering-Ergebnis haben.

4. Stemming der übrigen Wörter um Affixe und Suffixe zu entfernen. Dies ermöglicht einen Vergleich von Dokumenten mit syntaktischen Variationen der Indexterme.

Zur Veranschaulichung der Erstellung einer Repräsentation eines Textes sei hier Abbildung 4.2 angeführt. Die Abbildung zeigt den Prozess der Überleitung einer Volltextrepräsentation auf high-level Indexterme. Im folgenden werden die einzelnen Phasen 1 bis 4 im Detail erklärt und in eine geeignete Abfolge gebracht.

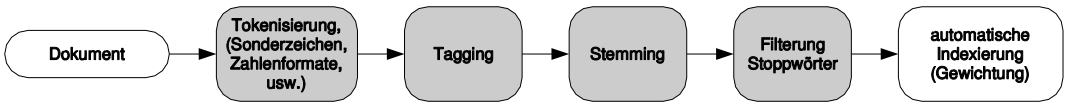



Abbildung 4.2: Textrepräsentation - Vom Volltext zu Indextermen

Nachdem diese Texttransformationen auf den Text angewandt wurden, folgt eine Auswahl der Terme (Stammformen), welche für den Index verwendet werden. Generell kann hierzu die syntaktische Eigenschaft eines Wortes herangezogen werden. Allgemein gesprochen überbringen hierbei Nomen mehr Bedeutung als Adjektive, Adverbien oder Verben, worauf in den Kapiteln 4.3 und 4.4 näher eingegangen wird.

### 4.2.2 Tokenisierung (Tokenizing)

Die lexikalische Analyse wandelt einen Fluss von Zeichen (den Text des Dokuments) in einen Strom von Wörtern um. Alle Wörter stellen grundsätzliche Kandidaten für Indexterme dar. Auf den ersten Blick erscheint es so, dass sich diese Aufgabe auf das Erkennen von Leerzeichen als Worttrennungszeichen beschränkt. Doch es bedarf mehrer Dinge als lediglich das Auffinden von Wortgrenzen. Andere Zeichen wie Zahlen, Bindestriche, Satzzeichen und Groß- und Kleinschreibung bilden einen nicht unbeträchtlichen Teil eines Textes und müssen somit berücksichtigt werden [4].

Zahlen sind generell keine guten Indexterme, da diese im Speziellen sehr kontextabhängig sind. Beispielsweise könnte ein Benutzer an der Zahl der Autounfälle zwischen 1990 und 2000 interessiert sein. Eine solche Abfrage in Bezug auf Indexterme könnte folgendermaßen aussehen: „Zahl + Autounfälle + 1990 + 2000“. Allerdings können die Zahlen 1990 und 2000 zu einem Retrieval von vielen unrelevanten Texten führen, die sich ebenfalls auf diesen Zeitraum beschränken. Das Problem dabei besteht darin, dass Zahlen ohne den Kontext sehr vage sind. Oftmals treten aber Zahlen in Kombination mit Buchstaben auf, beispielsweise bei „14tägig“ oder „100fach“, die sehr wohl aussagekräftig sind. In einem solchen Fall ist nicht klar, welche Regel nun angewandt werden soll. Allerdings kann auch eine Reihe von 16 Zahlen, die unter Umständen eine Kreditkartennummer identifiziert, von großem Interesse sein und muss ebenfalls bedacht werden. Eine mögliche Methode zur allgemeinen Behand-

lung von Zahlen besteht darin, Zahlen und Zahlenreihen vorerst zu ignorieren, es sei denn diese sind explizit (etwa durch Regular Expressions) näher bestimmt. Weiters ist es auch möglich, durch eine erweiterte lexikalische Analyse bestimmte Datums-, Zeit- oder andere Zahlenformate in einheitliche Darstellungsformen umzuwandeln.

Bindestriche stellen eine weitere Schwierigkeit bei der Analyse dar. Eine Möglichkeit stellt die Aufspaltung der durch Bindestriche getrennten Wörter dar. Beispielsweise kann „rot-weiss-rot“ oder „Objekt-orientiert“ aufgebrochen werden in „rot weiss rot“ und „Objekt orientiert“. Eine solche Aufspaltung ist jedenfalls mit Vorsicht durchzuführen, da sich dadurch auch die Semantik einer Aussage verändern kann.

Jedoch existieren auch Wörter die Bindestriche beinhalten, die nicht aufgebrochen werden können, beispielsweise „F-114“ oder „B-49“. Wiederum ist hier eine generelle Möglichkeit sinnvoll, wobei spezielle Fälle ebenfalls gesondert (etwa durch Regular Expressions) behandelt werden sollten.

Normalerweise werden Satzzeichen während des Prozesses der lexikalischen Analyse komplett entfernt. Einige Punktationen sind aber ein integraler Bestandteil von Wörtern, beispielsweise bei „etc.“ oder „Hr.“. Da solche Fälle allerdings minimal auftreten bzw. das Risiko einer Missinterpretation minimal ist, können diese vernachlässigt werden. Meistens handelt es sich hierbei um Abkürzungen, die bereits im Vorfeld durch den Einsatz von Textersetzern behandelt werden können. Da sowohl eine gestellte Abfrage als auch die einzelnen Dokumente denselben Analyseschritten folgen, wird das Ergebnis des Retrievals dennoch nur beschränkt beeinflußt. Einige spezielle Ausnahmen können auch hier Zusatzregeln sinnvoll machen. Sollte zum Beispiel Quellcode direkt in den Text eingebunden sein, so kann eine Unterscheidung der Variablen „x.id“ von „xid“ sinnvoll sein. Der Punkt sollte hier ebenfalls nicht verworfen werden.

Die Groß- und Kleinschreibung ist üblicherweise nicht wichtig für die Identifikation von Indexwörtern. Aus diesem Grund konvertieren Tokenizer im Normalfall den gesamten Text entweder in Klein- oder Großbuchstaben. Allerdings gibt es wiederum einige Szenarien, in denen eine Berücksichtigung von Interesse ist. Beispielsweise kann eine Abfrage bezüglich der Kommandosprache von Unix ein explizites nicht Umwandeln wünschenswert machen, da dies ein fixer Bestandteil des Unix-

Sprachgebrauchs ist. Durch eine Buchstabenumwandlung wird also Semantik verloren. Gerade in der deutschen Sprache zeichnet die Großschreibung grundsätzlich Nomen und Eigennamen aus, die bei der Ermittlung der Indexterme von besonderen Interesse sind. Aber auch andere Wörter wie „Lauf“ und „lauf“ haben, wie viele andere Wortpaare auch, eine unterschiedliche Bedeutung. Deshalb ist ein Inbetrachtziehen dieser (morpho-syntaktischen) Eigenschaften von Wörtern für das Deutsche ebenfalls von Interesse.

Alle vorgestellten Textoperationen können ohne große Schwierigkeiten implementiert werden. Gerade deshalb sollte hier vorsichtig vorgegangen werden, da diese Analyseergebnisse als erster Schritt der Retrieval Kette das Ergebnis sehr stark beeinflussen, da alle weiteren Schritte darauf aufbauen.

Dies ist natürlich in solchen Situationen ungünstig, in denen der Benutzer den genauen Vorgang der Indexerstellung nicht nachvollziehen kann. Leider gibt es dafür keine eindeutige Lösung. Wie schon gesagt verzichten einige Web-Suchmaschinen völlig auf Textoperationen, da sich dadurch die Interpretation einfacher gestaltet. Welche dieser Strategien nun zukunftsträchtiger ist, bleibt abzuwarten.

### 4.2.3 Identifikation von Wortkategorien (Tagging)

In einer Volltext-Repräsentation von Texten werden alle Wörter als Indexterme verwendet. Als Alternative kann eine abstraktere Sichtweise herangezogen werden, die nicht alle Wörter als Indexterme verwendet. Dies bedeutet, dass ein automatisches Auswahlverfahren angewandt werden muss, für das verschiedene Techniken eingesetzt werden können.

Ein guter Ansatz stellt die Identifikation von Nominalgruppen dar. Ein Satz in einem natürlichsprachlichen Text besteht aus Nomen, Pronomen, Artikeln, Verben, Adjektiven, Adverbien und Bindewörtern. Die Positionierung von Worten verschiedener grammatikalischer Kategorien findet innerhalb eines Satzes meist bewusst statt, um bestimmte Sachverhalte zu betonen. Dennoch kann argumentiert werden, dass die meiste Semantik über die Kategorie der Nomen transportiert wird. Deshalb ist die Auswahl von Nomen zur Bildung eines Indexes eine intuitive und vielversprechende

Strategie. Dies kann durch eine systematische Ausfilterung aller anderen Wortkategorien bewerkstelligt werden.

Da sehr oft mehrere Nomen zu einer eigenständigen Komponente verbunden werden („Institut für Wirtschaftswissenschaften"), kann es als sinnvoll erachtet werden, mehrere Nomen, die nahe beieinander im Text auftreten, als einen einzelnen Indexeintrag zu behandeln. Anstelle eines einfachen Nominalindexes spricht man von einer Indexierung mittels Nominalgruppen. Eine solche Nominalgruppe kann als eine Menge von Nomen, deren syntaktische Distanz (gemessen an der Menge des Strings dazwischen) einen vordefinierten Grenzwert (beispielsweise 3) nicht überschreitet. Werden Nominalgruppen zur Indexierung eingesetzt, wird eine konzeptuelle Sicht der Dokumente hinsichtlich einer Menge nicht-einelementiger Indexterme angewandt.

Um den Wörtern eines Textes ihre syntaktischen Wortkategorien zuweisen zu können, wird wiederum ein Regelwerk benötigt. Der bekannte Brill Tagger [9, 10, 8] verwendet beispielsweise drei Methoden, um die Wortkategorie eines Wortes festzustellen:

1. Ein Lexikons zur Bestimmung/Einschränkung der Wortkategorie
2. Eine Menge von Wortbildungsregeln (z.B. $'*ung' \longrightarrow' Nomen'$)
3. Ein Regelwerk zur kontextbezogenen Kategorienbestimmung

Am Ende des Vorgangs wird jedem Wort die wahrscheinlichste Wortkategorie zugewiesen.

### 4.2.4 Wortnormalisierung (Stemming)

Sehr oft kommen in einem Dokument Wörter vor, die im eigentlichen Text anderer Dokumente lediglich als eine Variante dieses Wortes aufscheinen. Mehrzahl, Gerundive oder verschiedene Zeitformen von Verben sind Beispiele für solche syntaktischen Variationen, die ein exaktes Übereinstimmen zwischen Indextermen verschiedener Dokumente verhindern. Dieses Problem kann durch ein Ersetzen dieser Terme durch ihre Stammform zumindest teilweise vermieden werden.

Die Stammform eines Wortes kann definiert werden als all jenes, das übrigbleibt, wenn einem Wort alle Affixe (Präfixe und Suffixe) entfernt werden. Ein typisches Beispiel einer solchen Stammform ist „arbeit“, das den Wörtern „arbeiten“, „arbeitet“, „Arbeit“, „Arbeiter“, „gearbeitet“, usw. zugrundeliegt. So können diese Stammformbildungen zu einer Verbesserung der Retrieval-Performanz eingesetzt werden, da sie Wortvarianten desselben Ursprungs auf ein zugrundeliegendes Konzept zusammenführen. Ein zweiter Effekt des Stemmings ist die Reduktion des Vokabulars, das auf die unterschiedlichen Stammformen abgebildet wird.

Trotz der Vorteile des Einsatzes von Stemmern herrscht in der Literatur keine Einigkeit darüber, ob Stemming wirklich eine Verbesserung des Retrievals darstellt (verschiedene Studien kamen zu verschiedenen Ergebnissen). Frakes [24] verglich acht Studien über die Vor- und Nachteile des Stemmings. Obwohl er ein Befürworter der Wortnormalisierung ist, erlaubt sein Ergebnis keine eindeutige Antwort. Aus diesem Grund verzichten auch einige Web-Suchmaschinen auf den Einsatz von Stemmern.

Frakes [24] unterscheidet vier Arten von Stemming Strategien: Entfernung von Affixen, Table Lookup, Successor Variety und n-Gramme. Table Lookup besteht darin, aus einer Tabelle die Stammform eines Wortes zu suchen. Es ist ein sehr einfaches Verfahren, aber die Stammformdaten aller Wörter einer Sprache benötigt. Da solche geschlossenen Daten allerdings noch nicht verfügbar sind und wahrscheinlich beträchtlichen Speicherplatz bedürfen, kommt dieser Art des Stemmings noch keine praktische Bedeutung zu. Successor Variety basiert auf der Bestimmung von Morphemgrenzen und verwendet dazu Informationen aus dem Bereich der Strukturlinguistik. Dies ist weit aufwendiger und komplexer als Affixentfernung. Die Strategie von n-Grammen basiert auf der Identifikation von Digrammen und Trigrammen und entspricht so mehr einem Termclustering als einem Stemming Algorithmus. Von allen Arten ist die Affixentfernung am intuitivsten, einfachsten und kann effizient implementiert werden, weshalb sich die weitere Diskussion auf diese Variante des Stemmings konzentriert.

Bei der Entfernung von Affixen kommt den Suffixen (im Gegensatz zu Präfixen) eine wesentlichere Bedeutung zu, da die meisten Wortvarianten durch ein hinzuziehen eines Suffixes zu ihrer Stammform gebildet werden. Einer der bekanntesten Stemmer

dieser Strategie stellt der Stemming Algorithmus von Porter dar. Die dahinterstehende Grundidee beruht auf einem Ersetzungsregelwerk von Suffixen. Beispielsweise wird die Regel

$n \longrightarrow null$

verwendet, um Pluralformen auf den Singular umzuwandeln, z.B.:

$Blumen \longrightarrow Blume$
$Strassen \longrightarrow Strasse$

Dabei bezeichnen Suffixe die letzten Buchstaben von Wörtern. Wichtig dabei ist, Regeln mit längeren Ersetzungsfolgen zuerst anzuwenden. Ein Beispiel ist

$ungen \longrightarrow null$
vor
$n \longrightarrow null$

So wird das Wort „Bedeutungen“ auf „Bedeut“ zurückgeführt, und nicht auf „Bedeutunge“. Der Stemming-Algorithmus nach Porter teilt den Ablauf dieses Regelwerks in fünf verschiedenen Phasen ein, wodurch er effektiv und schnell abgearbeitet wird. Näheres dazu findet sich in [4].

### 4.2.5 Filterung (Stopwortlisten)

Wie schon in Kapitel 4.2.1 besprochen, sind Wörter die sehr häufig in vielen Texten auftreten, keine guten Kandidaten für eine Indexierung, da diese nur sehr beschränkt zur Unterscheidung von Dokumenten beitragen. Zum Beispiel ist ein Term der in 80 Prozent aller Dokumente einer Kollektion vorkommt, nicht besonders wertvoll für die Bildung eines Indexes. Diese oft auftretenden Wörter nennt man kurzerhand Stopwörter, die generell aus der Liste der potentiellen Indexterme ausgefiltert werden. Artikel, Präpositionen und Konjunktionen werden naturgemäß als solche Stopwörter angesehen.

Die Elliminierung von Stopwörtern hat einen zusätzlichen Vorteil, da sie die Größe der Indexstruktur beträchtlich reduziert. Im Durchschnitt kann auf diese Weise eine Verringerung des Vokabulars auf bis zu 40 Prozent der ursprünglichen Größe erreicht werden [4].

Da durch Stopwortlisten der Index beträchtlich verkleinert werden kann, können ebenfalls auch andere Wörter als Artikel, Präpositionen und Konjunktionen als Stopwörter in Betracht gezogen werden. Beispielsweise können verschiedene Nomen, Verben, Adverbien oder Adjektive ebenfalls als Stopwörter behandelt werden. Dies ist besonders von der Domäne abhängig, in der ein solches System zum Einsatz kommt. Auf diese Weise kann, zusätzlich zu den semantikfreien Termen, fachspezifischer Wortschatz ohne Eigensemantik ausgefiltert werden. Beispielsweise ist der bereits erwähnte Term „Computer“ im Fachgebiet der Informatik kein besonders aussagekräftiger Diskriminator.

Neben diesen Vorteilen kann die Elliminierung von Stopwörtern zu einer Verminderung des Recalls führen. Sollte eine Abfrage „sein oder nicht sein“ abgesetzt werden, kann die Eliminierung von Stopwörtern den Inhalt auf „sein“ reduzieren. Dies macht ein Erkennen der gewünschten Phrase in einem Text fast unmöglich und stellt einen weiteren Grund dar, warum Web-Suchmaschinen sich oftmals auf eine reine Volltextsuche beschränken.

## 4.3 Gewichtung mittels des Vektorenmodells (Vector Space Model, VSM)

Das Vektorenmodell ist das am häufigsten verwendete Modell für die Aufgaben des IRs. Der Vorteil, im Gegensatz zu anderen Modellen wie dem Booleanmodell[1], ist das teilweise Übereinstimmen von Dokumenten (oder einer Abfrage und Dokumenten). Dies wird durch nichtbinäre Gewichtungen von Indextermen ermöglicht. Diese Gewichtungen werden anschließend dazu verwendet ein Ähnlichkeitsmaß zwischen Repräsentationen von Dokumenten (oder einer Abfrage und einem Dokument) zu errechnen. Durch eine absteigende Sortierung dieser Ähnlichkeitsmaße werden vom System auch Dokumente herangezogen, die einer Abfrage nur teilweise entsprechen. Das Ergebnis des gesamten Prozesses der so gereihten Dokumente ist meist besser (in Bezug auf die Bedürfnisse des Benutzers) als bei einem Retrieval von Dokumenten,

[1] Im Booleanmodell [4] wird ein Dokument mittels Termen repräsentiert. Ein Term ist entweder Teil der Repräsentation (Termgewicht $w_i = 1$) oder nicht (Termgewicht $w_i = 0$).

die entweder vollständig mit einer Abfrage korrelieren oder überhaupt nicht.

**Definition 2 – Vektorenmodell [4]**

Das Vektorenmodell verwendet Werte zwischen 0 und 1 um Paare der Form (Schlüsselwort $k_i$, Dokument $d_j$) zu gewichten (siehe Definition 3). Dies wird sowohl auf eine Abfrage als auch auf die zu vergleichenden Dokumente angewandt. Sei $w_{i,q}$ definiert als das Gewicht des Paares $[k_i, q]$, wobei $w_{i,q} \geqq 0$ ist. Dann ist der resultierende Abfragevektor $\overrightarrow{q} = (w_{1,q}, w_{2,q}, w_{3,q}, \ldots, w_{t,q})$, wobei $t$ die Gesamtzahl der verschiedenen Indexterme (das Vokabular) und $q$ die Repräsentation der Abfrage ist. Wie zuvor wird das zu vergleichende Dokument gleich behandelt, wobei der Dokumentvektor die Form $\overrightarrow{d_j} = (w_{1,j}, w_{2,j}, w_{3,j}, \ldots, w_{t,j})$ hat.

Es wird also das Dokument $d$ und die Abfrage $q$ als t-dimensionaler Vektor $\overrightarrow{d}$ und $\overrightarrow{q}$ repräsentiert (siehe Abbildung 4.3). Im Vektorenmodell wird nun die Korrelation dieser beiden Vektoren $\overrightarrow{q}$ und $\overrightarrow{d_j}$ verwendet, um ihre Ähnlichkeit auszudrücken. Diese Korrelation kann beispielsweise durch die Quantifizierung des Kosinus des Winkels $\alpha$ zwischen diesen Vektoren ausgedrückt werden.

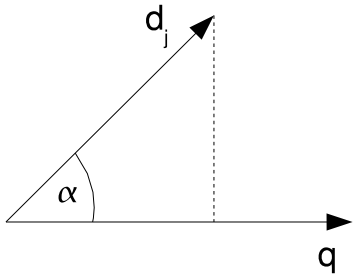



Abbildung 4.3: Ähnlichkeitsberechnung, aus [4, Seite 28]

Dies kann mittels der Formel

$$sim(d_j, q) = \frac{\overrightarrow{d_j} \bullet \overrightarrow{q}}{|\overrightarrow{d_j}| \times |\overrightarrow{q}|} = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{j=1}^{t} w_{i,q}^2}} \tag{4.2}$$

errechnet werden, wobei $|\overrightarrow{d}|$ und $|\overrightarrow{q}|$ die Normen (senkrechte Abbildung) des Dokument- und Abfragevektors darstellen. Der Faktor $|\overrightarrow{q}|$ im Nenner der Gleichung wirkt sich

dabei nicht auf die Berechnung und die anschließende Reihung aus, da er für jedes Dokument gleich ist. Der Faktor $|\overrightarrow{d}|$ dient der Normalisierung des Dokumentraumes.

Da $w_{i,j} \geqq 0$ und $w_{i,q} \geqq 0$ sind, variiert die Ähnlichkeit $sim(q, d_j)$ im Bereich zwischen 0 und +1. Im Gegensatz zu einer Entscheidung, ob ein Dokument nun relevant ist oder nicht, drückt das Vektorenmodell ein Maß der Ähnlichkeit (in Bezug eines Dokumentes mit einer Abfrage) aus. Ein Dokument kann also auch bezogen werden, indem es einer Abfrage nur teilweise genügt. Auf diese Art und Weise werden alle vorhandenen Dokumente gereiht an den Benutzer zurückgeliefert. Um unrelevante Ergebnisse auszuschließen, kann ein Grenzwert für die Ähnlichkeit festgelegt werden, der überschritten werden muss. Um diese Ähnlichkeitsreihung zu errechnen, muss zuerst die Bestimmung der Termgewichte genau festgelegt werden.

Termgewichtungen können auf verschiedene Weise bestimmt werden. Einen guten Überblick über die verschiedenen Möglichkeiten bietet Salton and McGill [75]. Diese Arbeit konzentriert sich auf die wesentlichen Merkmale dieser Methoden, die den effektivsten Termgewichtungstechniken zugrundeliegen.

Sei eine Kollektion $C$ von Objekten und eine vage Beschreibung einer Menge $A$ gegeben, so kann das Ziel eines einfachen Clustering-Algorithmus (lokales Clustering, siehe Abbildung 2.5 auf Seite 19) darin bestehen, diese Kollektion in zwei Mengen einzuteilen: Eine erste Menge von Objekten, die ähnliche Dokumente zu $A$ enthält, und eine zweite Menge, die sich aus nicht verwandten Objekten der Menge $A$ zusammensetzt. Eine vage Beschreibung meint hier, das keine vollständigen Informationen darüber vorliegen, ob ein Objekt nun der Menge $A$ angehört oder nicht. Als Beispiel kann eine Menge $A$ von Autos angenommen werden, deren Preis vergleichbar mit dem eines Porsche ist. Da die Beschreibung „vergleichbar“ nicht exakt gefasst werden kann, liegt keine genaue (und eindeutige) Beschreibung der Menge $A$ vor. Intelligentere Clustering-Algorithmen teilen Kollektionen aufgrund ihrer Dokumenteigenschaften in mehrere Cluster. Anschließend kann eine Ähnlichkeit zu einem (oder mehreren) Clustern auf die gleiche Weise wie zwischen Dokumentrepräsentationen ermittelt werden.

Laut Salton [75] kann das Problem des IR als ein Problem des Clusterings gesehen werden. Betrachtet man eine Kollektion $C$ von Dokumenten und eine Benutzerabfra-

ge als eine vage Beschreibung einer Menge $A$ von Objekten, so kann die IR Aufgabe reduziert werden auf ein Feststellen, welche Dokumente in dieser Menge $A$ liegen und welche nicht. Das IR Problem kann also als ein Clusteringproblem aufgefasst werden.

Beim Clustering müssen zwei Hauptfragen geklärt werden. Als erstes müssen die Eigenschaften bestimmt werden, die die Objekte der Menge $A$ bestmöglich beschreiben. Zweitens muss festgelegt werden, welche Eigenschaften die Objekte der Menge $A$ von den verbleibenden Objekten der Kollektion bestmöglich unterscheiden. Die erste Menge von Eigenschaften beschreibt eine Quantifizierung der sogenannten intra-cluster Ähnlichkeit, während die zweite Menge von Eigenschaften eine Quantifizierung der inter-cluster Unähnlichkeit ausdrückt. Die erfolgreichsten Clustering-Algorithmen versuchen beide dieser Effekte auszugleichen.

Beim Vektorenmodell wird die intra-cluster Ähnlichkeit durch die relative Auftrittshäufigkeit eines Terms $k_i$ in einem Dokument $d_j$ gemessen, wodurch jeder Term in die Berechnung mit einfließt. Diese Termfrequenz (term frequency) wird meist als $tf$-Faktor bezeichnet und liefert Informationen darüber, wie gut ein Term den Dokumentinhalt beschreibt. Im Gegensatz dazu wird die inter-clustering Unähnlichkeit durch eine Messung der inversen Frequenz eines Terms $k_i$ über allen Dokumenten der Kollektion angegeben. In der Literatur wird dies als inverse Dokumentfrequenz (inverse document frequency), oder kurz als $idf$-Faktor, bezeichnet. Die Motivation für den Einsatz der inversen Dokumentfrequenz eines Terms ist die Idee, dass in jedem Dokument auftretende Terme nicht sehr hilfreich bei der Unterscheidung relevanter und unrelevanter Dokumente sind. Wie auch bei guten Clustering-Algorithmen verwenden vielversprechende Gewichtungsschemata im IR eine Ausgewogenheit dieser beiden Effekte.

Verschiedene Variationen dieses Gewichtungsschemas finden sich bei Salton und Buckley [76]. Generell betrachtet, liefert die Form der Termgewichtung (siehe 4.3) gute Resultate in vielen Kollektionen [4].

**Definition 3 – Termgewichtung**

Sei $N$ die Gesamtanzahl der Dokumente eines Systems und $n_i$ die Anzahl der Dokumente, in denen der Term $k_i$ auftritt. Sei weiters $freq_{i,j}$ die Auftrittshäufigkeit eines Terms $k_i$ im Dokument $d_j$ (also die Anzahl des Vorkommens vom Term $k_i$ im Dokument $d_j$). Dann kann die normalisierte Frequenz $f_{i,j}$ des Terms $k_i$ im Dokument $d_j$ angegeben werden als

$$f_{i,j} = \frac{freq_{i,j}}{max_l freq_{l,j}} \tag{4.3}$$

wobei das Maximum über allen Termen des Dokuments $d_j$ errechnet wird. Wenn der Term $k_i$ nicht im Dokument $d_j$ auftritt, dann nimmt $f_{i,j}$ den Wert 0 an. Die inverse Dokumentfrequenz $idf_i$ für den Term $k_i$ kann angegeben werden als

$$idf_i = \log \frac{N}{n_i} \tag{4.4}$$

Die bekanntesten Schemata zur Termgewichtung haben die Form

$$w_{i,j} = f_{i,j} \times idf_i \tag{4.5}$$

oder diverse Variationen dieser Formel. Generell werden solche Gewichtungsstrategien als $tf.idf$ Schemate bezeichnet.

Um die Terme von Abfragen zu gewichten, schlägt Salton und Buckley folgende Variante vor:

$$w_{i,q} = \left(0.5 + \frac{0.5 freq_{i,q}}{max_l freq_{l,q}}\right) \times \log \frac{N}{n_i} \tag{4.6}$$

wobei $freq_{i,q}$ die Auftrittshäufigkeit des Terms $k_i$ im Text der Abfrage $q$ ist.

Die Hauptvorteile des Vektorenmodells sind laut [4]:

- Die Gewichtung der Terme verbessert die Performanz des Retrievals.
- Die teilweise Übereinstimmung von Repräsentationen erlaubt ein Retrieval von Dokumenten, die eine Abfragebedingung lediglich approximieren.

- Die Kosinus-Reihungsfunktion sortiert die bezogenen Dokumente anhand ihrer Ähnlichkeit.

Neben seiner Einfachheit bietet das Vektorenmodell eine stabile Reihungsstrategie bei vielen Kollektionen. Die erhaltenen Ergebnisse sind ohne Abfrageerweiterungen (query expansions) bzw. Benutzerfeedback (relevance feedback) innerhalb dieses Modells nur schwer zu verbessern. Eine Vielzahl anderer Berechnungsmethoden wurden vorgestellt und mit dem Vektorenmodell verglichen. Es stellte sich heraus, dass das Vektorenmodell entweder besser oder zumindest annähernd so gut ist wie andere Alternativen. Darüberhinaus ist das Vektorenmodell sehr einfach und effizient, weshalb es auch heute noch eines der populärsten Modelle im IR darstellt.

Da der Schreibstil von Personen sehr individuell ausfällt, reicht eine reine Gewichtung von Wörtern basierend auf ihrer Auftrittshäufigkeit oftmals nicht aus, um die Semantik von Dokumenten zu erfassen. Relationen zwischen Wörtern werden ungeachtet gelassen, der Beitrag selten auftretender Wörter zur Inhaltsbildung wird unterschätzt, und nicht zuletzt werden die Strukturen eines Dokuments nicht in die Analyse miteinbezogen [6].

Ein Nachteil des Vektorenmodells besteht gerade in der Annahme unabhängiger Indexterme (Gleichung 4.5 berücksichtigt keine Termabhängigkeiten). Deshalb wird dem Vektorenmodell nachgesagt, Textdokumente lediglich als Ansammlung von Wörtern (bag of words) zu interpretieren. Die Aufgabe der Analyse wird auf ein Reduzieren des Textes um vorhandene Stopwörter und eine anschliessende Wortzählung limitiert. Neuere Ansätze aus dem Bereich des NLP ergänzen diese Aufgaben hingegen um den Einsatz von Methoden, die Bereits im Kapitel 4.2 behandelt wurden.

Die Einbeziehung von Termabhängigkeiten in die Repräsentation von Dokumenten einer Kollektion kann jedoch auch einige Probleme mit sich bringen. Da solche Abhängigkeiten meist nur lokaler Natur sind, kann sich dessen Einbeziehung auf die gesamte Kollektion nachteilig für das gesamte Retrievalergebnis auswirken und sollte deshalb mit Vorsicht genossen werden [4].

## 4.4 Gewichtung mittels des Thema-Rhema Modells

Das zuvor vorgestellte Vektorenmodell stellt einen Ausgangspukt zur Erstellung von Dokumentrepräsentationen dar. Es werden jedoch andere wichtige Texteigenschaften wie die Textorganisation (text organization), der Textzusammenhang oder Textfluss (text connectivity) und die Satzstruktur (sentence structure) nicht behandelt. Um diese Aspekte ebenfalls einfließen zu lassen, ist eine genauere Analyse der Dokumente notwendig.

Im Gegensatz zu einer tiefen linguistischen Satzanalyse, die etwa beim Einsatz von NLP Techniken auf verschiedenen Stufen (lexikalisch, syntaktisch, semantisch, pragmatisch) durchgeführt wird, ist eine oberflächliche Analyse oft ausreichend [6]. Eine volle Analyse der natürlichen Sprache ist fokussierter als es beim Überfliegen oder Durchsuchen von Texten erforderlich ist. Überfliegen bedeutet in diesen Zusammenhang das grobe inhaltliche Erfassen von Texten, indem hauptsächlich der erste und letzte Satz von Absätzen gelesen und nach Zusammenfassungen gesucht wird. Durchsuchen hingegen beschreibt den Vorgang der Suche nach spezifischen Sachverhalten im Text wie Schlüsselwörter oder Phrasen.

In dieser Arbeit wird die Textanalyse als ein Aufbrechen des Textes in einzelne Komponenten verstanden. Dies wird durch ein Betrachten der verschiedenen Aspekte von Texten erreicht. Eine linguistische Theorie, die Systemic Functional Theory (SFT), dient dabei als Ausgangspunkt. Die Thema-Rhema Theorie, ein Teilbereich der SFT, stellt die Grundlage der hier vorliegenden Arbeit dar und wird genauer behandelt.

Die SFT geht von einem zugrundeliegenden funktionalen Modell von Sprachsystemen aus. Dabei wird die Sprache als ein Mittel zum Ausdruck von Bedeutung gesehen. Dieses System besteht aus verschiedenen Komponenten und Abstraktionsstufen, die funktional miteinander verknüpft sind, wobei jede Komponente eine eigene Funktion, die zum Ausdruck von Bedeutung dient, innehat. In SFT werden die Begriffe „System“ und „Funktion“ verwendet, um Sprachen zu modellieren. Genauer gesagt werden diese Begriffe im Zusammenhang mit dem Ausdruck von Bedeutung verknüpft, deshalb auch der Name Systemic Funktional Linguistics Theory. Der Be-

griff der „Funktion“ steht laut Halliday [33] im Zusammenhang mit Sprachgebrauch (language use), Textstruktur (text structure) und Textzusammenhang (connectivity).

Der Begriff Sprachgebrauch muss nicht weiter geklärt werden, da die Funktion der Sprache Kommunikation ist. Dieser Informationsaustausch zwischen Menschen kann in verschiedenster Weise erfolgen (visuell, akustisch, schriftlich, sprachlich, pantomimisch, ...) und wird in natürlichsprachlicher Grammatik ausgedrückt.

Die Textstruktur beschreibt die Sprache als ein System bestehend aus einer Menge von Komponenten. Jede Komponente trägt einen bestimmten Typ von Bedeutung und steuert zur Bildung der Gesamtbedeutung einer Nachricht bei. Alle Sprachen sind um solche Bedeutungstypen organisiert, die als Metafunktionen bekannt sind. Jede Metafunktion gibt Auskunft über die Kernaussagen eines Textes.

Der Textzusammenhang steht in Verbindung mit Textelementen, den natürlichen Konstituenten einer Nachricht (Satz, Satzglied, ...), die die Metafunktionen bilden. Genaueres dazu findet man in [61].

Insgesamt gesehen, zielt die SFT auf die Beantwortung einer Frage ab: Wie strukturieren Menschen ihre Alltagssprache? Wenn man diesen Vorgang versteht, ist es einfach, eine Nachricht zu verstehen. Textverständnis entsteht somit durch eine Extraktion der Funktionen von Textelementen [6].

Zuvor wurde der Begriff der natürlichsprachlichen Grammatik verwendet, jedoch nicht weiter erklärt. Die traditionelle Grammatik endet am Satzende, wobei jedes Element eine einzelne Funktion übernimmt. In der SFT werden Sätze als konstruktive Sätze gesehen, in denen eine natürliche Konfiguration von Elementen innerhalb eines Satzes stattfindet. Jedem Element kommt hierbei eine bestimmte Funktion zu. Das bedeutet, das die meisten Elemente einer grammatikalischen Struktur multifunktional sind. Die SFT verwendet die traditionelle Sicht der Satzgrenzen, analysiert jedoch die einzelnen Konstituenten multifunktional [33].

Die SFT unterscheidet verschiedene Bedeutungen von Sätzen. Dies geschieht auf drei verschiedenen Ebenen, der zwischenmenschlichen (interpersonal), ideellen (ideational, experiential) und inhaltlichen (textual).

Die zwischenmenschliche Bedeutung ergibt sich aus der Darstellung der Sprecherrolle und der Einstellung des Autors zu einem gewissen Thema. Es handelt sich um die Semantik der Interaktion.

Mit der ideellen Bedeutung ist die Auffassung der Sprache bezüglich „jemand oder etwas tut etwas aus einem bestimmten Grund“ gemeint. Es handelt sich um die Vorstellungen und Erfahrungen des Schreibers.

Während die zwischenmenschliche und die ideelle Bedeutung mehr mit der lokalen Bedeutung von Sätzen zu tun hat, behandelt die inhaltliche Bedeutung die Art und Weise der Textorganisiation, also den Text als ein zusammenhängendes Stück in seiner Gesamtheit. Dies spiegelt sich in der Art der Sortierung und Zusammenführung der einzelnen Textbausteine wider. Es handelt sich um die Darstellung von Inhalt und wie dieser mit dem vorhergegangenen Inhalt in Verbindung steht. Die Frage der Textorganisation ist also entscheidend für die Entwicklung eines durchgängigen Konzepts in einem Text.

Da die Analyseeinheit innerhalb der SFT der Satz ist, basiert die inhaltliche Bedeutung auf der Annahme, das jeder Satz eine eigenständige Nachricht transportiert. Die Thema-Rhema Theorie geht deshalb davon aus, das jeder Satz in zwei Teile zerlegt werden kann, in einen *thematischen* und einen *rhematischen* Teil.

Das Thema (bzw. die Themata) eines Satzes kann folgendermaßen definiert werden: Thema eines Satzes ist das, wovon der Satz handelt (worum es inhaltlich geht). Es ist der Ausgangspunkt für den Autor für das, was noch gesagt werden soll.

Analog dazu kann das Rhema (bzw. die Rhemata) definiert werden als all das, das nicht Thema des Satzes ist. Es ist der Teil des Satzes, in dem das Thema entwickelt wird.

Dazu folgende Beispiele:

| | | |
|---|---|---|
| (1) | `Der Lehrer` | `erzählt den Studenten eine Geschichte.` |
| | Thema | Rhema |
| (2) | `Eine Geschichte` | `erzählt der Lehrer den Studenten.` |

In Beispiel (1) erzählt der Schreiber über einen „erzählenden Lehrer“, wohingegen im Beispiel (2) der Schreiber über eine „erzählte Geschichte“ berichtet. Der

Informationsfluß der beiden Beispiele ist nicht identisch. Es werden zwar die gleichen Konstituenten verwendet, jedoch stecken zwei verschiedene Aussagen hinter den Sätzen. Die inhaltliche Bedeutung wird also durch die spezielle Reihung der einzelnen Konstituenten ausgedrückt.

Die Thema-Rhema Theorie geht davon aus, dass das Thema in der Regel zu Beginn eines Satzes steht und somit das dem Leser bereits bekannte Wissen darstellt. Es ist derjenige Teil, in dem der Autor weitere Informationen im Rest des Satzes einbringt. Laut Firbas [63] drückt das Thema jedoch nicht immer nur bekannte Informationen aus, sodass diese Generalisierung abgeschwächt werden muss.

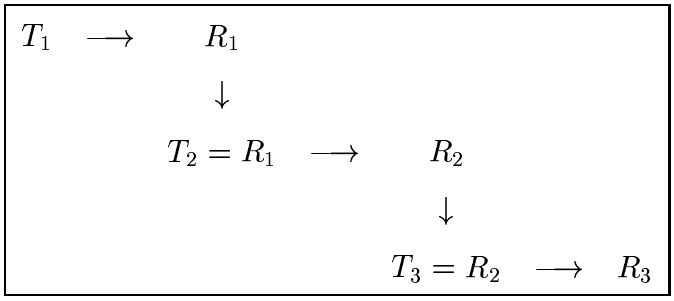
Im Gegensatz zum Thema beinhaltet das Rhema sehr wohl neue Informationen. Dies geschieht als eine Art Kommentar zu dem bereits im thematischen Teil behandelten Material. Neue Informationen werden generell im rhematischen Teil des Satzes vorgestellt, da Autoren dazu neigen, bedeutsame Informationen am Ende des Satzes zu Positionieren [25]. Im Gegenzug dazu wird der Satzanfang verwendet, um den Leser zur Nachricht im Rest des Satzes zu führen. Dieser Ansatz wird durch die Tendenz von Autoren bestätigt, die am Satzende an die bedeutsamen Fakten erinnern, bevor sie zum nächsten Satz übergehen. Generell ist das Auffinden neuer Informationen anhand von Positionen innerhalb von Sätzen keine leichte Aufgabe.

Die Textanalyse beschäftigt sich nicht nur mit der Analyse der einzelnen Sätze. Vielmehr wird der gesamte Text als eine Einheit betrachtet. Eine detaillierte Textanalyse führt zu NLP Techniken, die auf einer genauen Satzanalyse beruhen. Im Bereich des IR jedoch, das sich häufig mit einer großen Anzahl oftmals sehr umfangreicher Textdokumente auseinandersetzen muss, ist eine solche Analyse derzeit zu aufwendig.

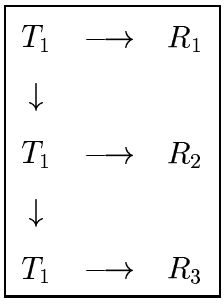
Die Verwendung der inhaltlichen Bedeutung stellt jedoch eine zuversichtliche Methode dar, um die Hauptthemen (topics) eines Textdokuments zu finden. Dabei nimmt sie Einfluß auf die Strukturierung und Organisation von Texten. Es wurden verschiedene Schemata entwickelt, nach denen die Themata in einem strukturierten Text entwickelt werden können [6]:

- Bei der **einfachen linearen Entwicklung** der Themata wird das rhematische Material im nächsten Satz wieder aufgefasst, um das neue Thema des Satzes

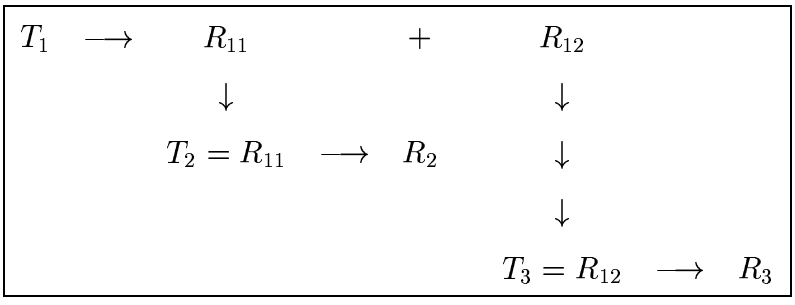
zu bilden. Es wird ein Textfluss gebildet, indem neue Information auf den kürzlich eingeführten Informationen aufbauen.




- Die **kontinuierliche (konstante) Entwicklung** der Themata verwendet in aufeinanderfolgenden Sätzen jeweils dasselbe Thema. Dabei unterliegen die einzelnen Rhemata keiner Reihenfolge. Das im Text genannte Thema wird in jedem Satz wiederholt, muss jedoch nicht zwingend immer dasselbe Wort verwenden (im Fall von Synonymität).




- Im Gegensatz dazu wird bei der **multiplen Entwicklung** der Themata ein komplexes Rhema bestehend aus vielen Informationen, zu einzelnen Themata überführt. In den nachfolgenden Sätzen werden die verschiedenen Teile des rhematischen Materials als Thema verwendet.




- Bei der **abgeleitete Entwicklung** der Themata sind Sätze auf ein sogenanntes Hyperthema bezogen, das nicht explizit im Text vorhanden ist. Um ein solches Hyperthema zu finden, muss externes Wissen (Weltwissen) hinzugezogen werden.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $[T]$ | | $=$ | $[T]$ | | $=$ | $[T]$ | | |
| $\downarrow$ | | | $\downarrow$ | | | $\downarrow$ | | |
| $T_1$ | $\longrightarrow$ | $R_2$ | $T_2$ | $\longrightarrow$ | $R_2$ | $T_3$ | $\longrightarrow$ | $R_3$ |

Studien von Nwogu [62] haben ergeben, dass die *einfache lineare* und die *kontinuierliche* Musterentwicklung eines Themas häufig auftreten. Die *einfache lineare Entwicklung* wird dabei häufiger in journalistischen Berichten verwendet, während die *kontinuierliche Entwicklung* häufiger in wissenschaftlichen Artikeln zu finden ist.

Aus diesen Entwicklungsmustern ist ersichtlich, dass Verfasser von Texten ihre Sätze in Hintergrund–Vordergrund Informationspaare gliedern. Dies ermöglicht dem Leser auf relevanten Informationen (Hintergrundwissen) aufzubauen, sobald neue Informationen (Vordergrundwissen) gebracht werden. Arbeiten von Halliday [33] und Fries [25] zeigen, das die wichtige Information eines Textes im thematischen Teil liegt. Das rhematische Material dient in der Regel als Zusatzinformation zum Thema.

### Einsatz der Thema-Rhema Theorie

Im Falle des Vektorenmodells wurde das Gewicht direkt aus der Auftrittshäufigkeit eines Terms berechnet. Mittels der Thema-Rhema Theorie kann ein anderer Weg der Gewichtung gewählt werden.

Die Dokumente werden in zwei Mengen von Indextermen aufgespalten, in Themata und Rhemata. Ebenfalls wird eine Relation „wird erklärt mit“ unter ihnen hergestellt. Die Gewichte der Themata entsprechen dem Einfluß dieser Konzepte auf die Bedeutung des Textes. Mit anderen Worten, das Gewicht zeigt den Beitrag eines Themas zur Repräsentation des Inhalts eines Textes.

Um die Anordnung, also thematisches Material mit erklärendem bzw. ausführendem rhematischen Material, darzustellen, kann eine $m \times n$ Matrix $M$ dienen. Die Zeileneinträge entsprechen dabei dem thematischen Material, die Spalteneinträge dem rhematischen Material eines Textes. Eine Zelle $r_{ij}$ dieser Matrix $M$ enthält eine Zahl die angibt, wie oft das Thema $T_i$ der Zeile $i$ zusammen mit dem Rhema $R_j$ der Spalte $j$ im Text auftritt. Die Menge der Themata und die Menge der Rhemata ist dabei unterschiedlich. Dennoch können dieselben Wörter in beiden Mengen auftreten.

Um eine Gewichtung der einzelnen thematischen und rhematischen Konzepte zu berechnen, werden zwei Metriken, das *Explanatory Power* und das *Topicality Power*, verwendet [6]. Die Erste gibt an, wie oft verschiedene Themata zusammen mit einem Rhema auftreten. Die Zweite berechnet, wie oft ein Thema durch verschiedene Rhemata näher ausgeführt wird. Durch diese Relationen zwischen den Indextermmengen kann das Topicality Power wie folgt definiert werden:

**Definition 4 – Topicality Power**

Das Topicality Power misst die Wichtigkeit eines Konzepts und ermöglicht das Auffinden der Hauptthemen in Dokumenten. Es wird berechnet durch die Aufsummierung der Auftrittshäufigkeiten des rhematischen Materials, die dieses Thema erklären.

$$TOP_i = \sum_j r_{ij} \tag{4.7}$$

In ähnlicher Weise reflektieren die Gewichte der Rhemata ihren Beitrag bei der Definition (nicht explizit genannter) Themata. So kann das Explanatory Power definiert werden als:

**Definition 5 – Explanatory Power**

Das Explanatory Power gibt den Beitrag eines rhematischen Konzepts zur Bildung eines Themas an. Es wird berechnet als Summe jedes Rhemas über alle Themata des gesamten Textes.

$$EXP_j = \sum_i r_{ij} \tag{4.8}$$

Als Beispiel sei hier folgender Text in Tabelle 4.1 angegeben. Die dazugehörige Thema-Rhema Matrix ist in Tabelle 4.2 angeführt.

Der Einfachheit halber wurden die einzelnen Wörter des Textes in Tabelle 4.2 nicht durch ihre Stammformen ersetzt. Wie die Inhalte der Tabelle 4.2 und die Kenngrößen

Tabelle 4.1: Thema-Rhema Beispiel

| | |
|---|---|
| (1) | `Ein Computer ist ein technisches Gerät und wird oft als PC bezeichnet.` |
| (2) | `Ein typischer PC besteht aus einer Platte, Speicher und einer CPU.` |
| (3) | `Die CPU übernimmt die mathematischen Rechnungen.` |
| (4) | `Die Festplatte speichert Daten auf lange Zeit.` |
| (5) | `Der Speicher dient der Programmausführung.` |
| (6) | `Der Speicher ist vielfach schneller als eine Festplatte.` |

Tabelle 4.2: Thema-Rhema Matrix

| | | **Rhemata** | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ist (1) | Gerät (1) | PC (1) | bezeichnet (1) | besteht (2) | Platte (2) | Speicher (2) | CPU (2) | übernimmt (3) | Rechnungen (3) | speichert (4) | Daten (4) | Zeit (4) | dient (5) | Programmausführung (5) | Festplatte (6) | **Topicality Power** |
| **Themata** | Computer (1) | 1 | 1 | 1 | 1 | | | | | | | | | | | | | **4** |
| | PC (2) | | | | | 1 | 1 | 1 | 1 | | | | | | | | | **4** |
| | CPU (3) | | | | | | | | | 1 | 1 | | | | | | | **2** |
| | Festplatte (4) | | | | | | | | | | | 1 | 1 | 1 | | | | **3** |
| | Speicher (5,6) | 1 | | | | | | | | | | | | | 1 | 1 | 1 | **4** |
| | **Explanatory Power** | **2** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | |

Topicality und Explanatory Power im Detail berechnet werden, wird im Kapitel 5 genau behandelt.

Unter Zuhilfenahme der Fuzzy-Set Theorie (siehe Anhang A) können Dokumente als Fuzzy-Sets repräsentiert werden. Jedem Element eines solchen Fuzzy-Sets entspricht dabei ein Indexterm mit einem jeweiligen Zugehörigkeitsgrad, der als Gewichtung

verwendet wird. Dieser Zugehörigkeitsgrad entspricht der Aussagekraft, die ein Term in Hinblick auf die Gesamtbedeutung des Textes hat.

Wie bereits erwähnt, ist die Relation „wird erklärt mit“ eine unscharfe Relation zwischen Themata und Rhemata. Formal kann ein thematisches Konzept als ein Fuzzy-Set wie folgt definiert werden:

**Definition 6 – Thematisches Konzept**

Die Repräsentation der Fuzzy Themata ist ein Fuzzy-Set $(R, h)$, wobei jedem Rhema aus $R$ über die Funktion $h : R \rightarrow [0, 1]$ einem jedem thematischen Konzept $T_i$ zugewiesen wird. Die Zugehörigkeitsfunktion $h$ weist jedem Rhema $R_j$ einen Zugehörigkeitswert $h_j$ der Form

$$h_j = \frac{\text{Anzahl Rhemata } R_j, \text{ die ein Thema } T_i \text{ erklären}}{\text{Topicality Power des Themas } T_i} \tag{4.9}$$

$$h_j = \frac{r_{ij}}{TOP_i} \tag{4.10}$$

zu.

Die Funktion $h$ entspricht einer lokalen Gewichtungsfunktion. Nachdem das Topicality Power eines Themas bestimmt wurde, berechnet $h$ den Beitrag (einen Wert zwischen 0 und 1) eines jeden Rhemas (des jeweiligen Themas) zum Topicality Power. Diese Berechnung stellt ebenfalls eine Normalisierung des Wertes dar.

Dokumente können als fuzzy Repräsentationen der enthaltenen Themata in Bezug auf alle Themata einer gesamten Kollektion gesehen werden. Der Zugehörigkeitsgrad eines Themas ist 0, wenn ein Thema in einem Dokument nicht vorkommt, beziehungsweise nimmt er einen Wert zwischen 0 und 1, der Auftrittshäufigkeit eines Themas normalisiert durch die Anzahl vorkommender Themata, an. Diese Zugehörigkeitsfunktion gleicht ebenfalls auch die verschiedenen Längen einzelner Dokumente durch eine Reduktion der Auftrittshäufigkeiten von Themata in relativ komplexen Dokumenten aus. Die thematische Komplexität wird nicht aus der Länge des Dokuments alleine berechnet, da der Einfluß von erklärenden Material ebenfalls betrachtet wird. Die thematische Komplexität wird berechnet durch das Vorhanden-

sein erklärenden Materials zu einem Thema im Verhältnis zur Menge aller Themata. Dies führt zu zwei weiteren Definitionen:

**Definition 7 – Thematische Komplexität**

Die thematische Komplexität ergibt sich aus der Summe aller vorhandenen Themata eines Dokuments.

$$C_T = \sum_i TOP_i = \sum_i \sum_j r_{ij} \tag{4.11}$$

**Definition 8 – Fuzzy Dokumentrepräsentation**

Eine fuzzy Dokumentrepräsentation ist ein Fuzzy-Set $(T, g)$ mit $g : T \rightarrow [0, 1]$, die jedem Dokument $D$ eine Menge von Themata $T$ zuweist. Die Zugehörigkeitsfunktion $g$ weist also jedem Thema $T_i$ einen Zugehörigkeitsgrad $g_i$ zu.

$$g_i = \frac{\text{Topicality Power des thematischen Konzepts } T_i \text{ in } D}{\text{Thematische Komplexitaet des Dokuments } D} \tag{4.12}$$

$$g_i = \frac{TOP_i}{C_T} \tag{4.13}$$

Die Definition beschreibt eine Dokumentrepräsentation nicht mehr als ein einzelnes Fuzzy-Set. Vielmehr handelt es sich um ein komplexes Fuzzy-Set, bei dem jedes einzelne Element wiederum ein Fuzzy-Set darstellt. Dies wird in der Literatur generell als Typ-2 Fuzzy-Set bezeichnet. Um diese Repräsentation und seine Nähe zu einer hierarchischen Struktur gerecht zu werden, wird hier der Begriff von „hierarchischen Fuzzy-Sets“ eingeführt [6]. Abbildung 4.4 veranschaulicht diese Hierarchie deutlicher.

Eine Kollektion von Dokumenten wird derart analysiert (siehe Abbildung 4.4): Im Vorfeld wird jedes einzelne Dokument separat analysiert, wobei die thematischen und rhematischen Elemente herausgearbeitet werden. Anschließend werden das Topicality und das Explanatory Power berechnet. Die Topicality Powers werden in

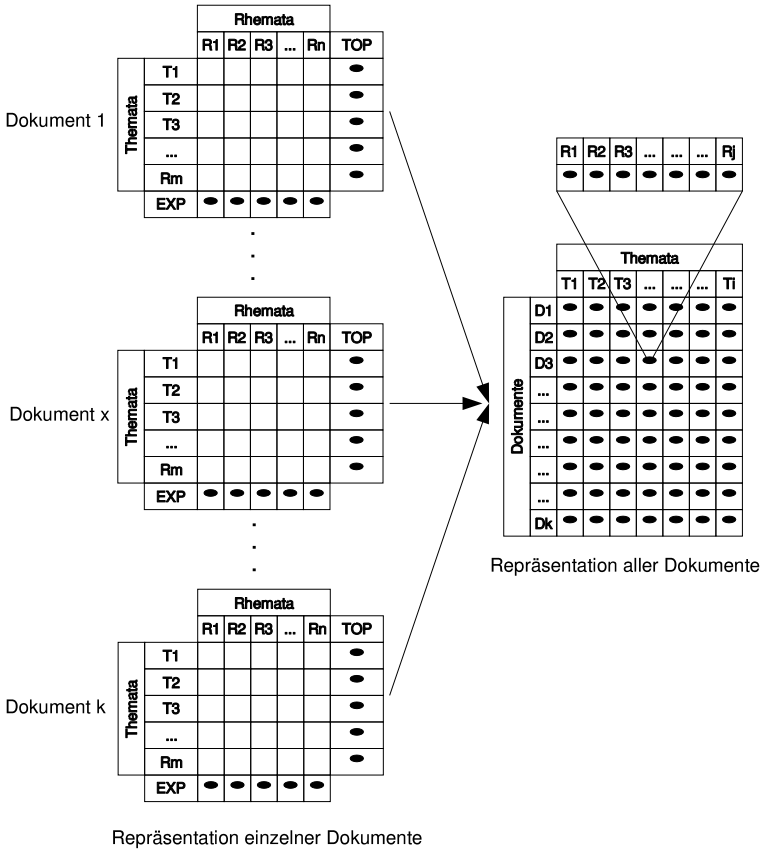



Abbildung 4.4: Dokumentrepräsentation des Thema-Rhema Modells

einer Matrix über den gesamten Dokumenten der Kollektion repräsentiert. Für jedes einzelne Topicality Power wird die Menge des rhematischen Beitrags gespeichert.

KAPITEL 5

# Das SyRS System - Der SyRS Prototyp

In den vorangegangenen Kapiteln wurde der theoretische Hintergrund für die Entwicklung eines Dokumentclustering-Systems aufgearbeitet. In diesem Kapitel wird der im Zuge dieser Arbeit verwendete und weiterentwickelte Prototyp SyRS und dessen Arbeitsweise beschrieben.

## 5.1 Architektur

Im Zuge seiner Dissertation [6] entwickelte Dr. Bouchachia den Information Retrieval Prototyp SyRS (**Sy**stemic **R**etrieval **S**ystem). Basierend auf seiner Arbeit zur Verarbeitung englischer Textdokumente wurde das System für das Deutsche adaptiert und erweitert. Sämtliche im System vorhandenen Komponenten wurden umgestaltet. Aufgabe des Systems ist es, ein Dokumentclustering vorzunehmen. Aus einer vorgegebenen Dokumentensammlung werden diejenigen Dokumente an den Benutzer zurückgeliefert, die aufgrund einer gestellten Abfrage inhaltlich von Interesse sind. Die Abfrage wird, im Gegensatz zu traditionellen IR Systemen, jedoch nicht mehr mit allen Dokumenten einzeln verglichen, sondern mit den Clusterzentren. Wenn ein Clusterzentrum mit einer Abfrage korrelliert, werden alle Dokumente dieses Clusters bezogen. Es wird also ein Clusterretrieval (siehe Kapitel 2.5) durchgeführt.

SyRS besteht aus zwei Hauptmoduln (siehe Abbildung 5.1): dem Natural Language Modul und dem Neuronalen Netzwerk Modul. Diese Haupteinheiten gliedern sich wiederum in selbstständige, serialisierte Komponenten. Dies ermöglicht den Aus-

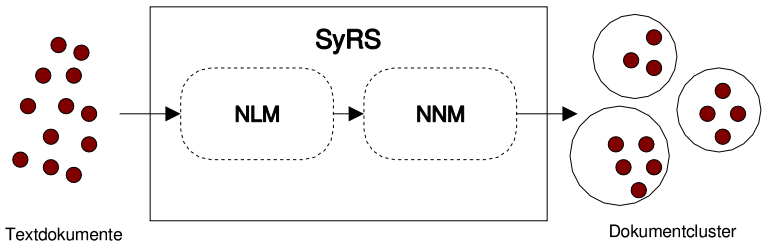



Abbildung 5.1: SyRS Übersichtsgrafik

tausch gesamter Funktionseinheiten, wodurch das System einfach zur Verarbeitung anderer Sprachen angepasst werden kann.

Der Datenaustausch zwischen diesen Komponenten erfolgt über Textdateien, wobei der Output einer Komponente als direkter Input der darauffolgenden Einheit weitergereicht wird. Die erste Komponente, der Tokenizer, erhält die Textdokumente als unformatierte Textdateien zur Verarbeitung. Die letzte Komponente, das fuzzy Adaptive Resonance Theory Netz, liefert die Clusterzuweisungen des jeweiligen Textdokuments zurück und speichert diese in einer Ergebnisdatei.

Im Folgenden werden alle Einzelkomponenten des Systems, deren Funktionalität, Implementierungsdetails und Aufrufhierarchie im Detail beschrieben.

## 5.2 Das Natural Language Modul

Das Natural Language Modul übernimmt die Aufgabe der gesamten Textaufbereitung und -analyse. Das Ergebnis stellt eine Repräsentation des Dokuments dar, die an das Neuronale Netzwerk Modul weitergegeben wird.

### 5.2.1 Aufbau

Das Natural Language Modul ist für die gesamte Textaufbereitung und linguistische Textanalyse zuständig. Die zu verarbeitenden (Roh-)Texte, die sowohl einzeln als auch im Batchmodus (über Dateiargumente) behandelt werden können, werden entsprechend aufbereitet und ihrer Eingabereihenfolge nach hintereinander verarbeitet.

Das Natural Language Modul selbst gliedert sich wiederum in fünf Einzelkomponenten (siehe Abbildung 5.2):

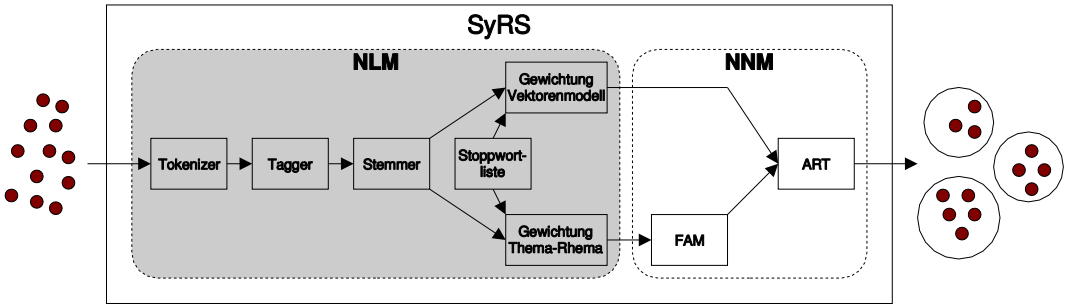



Abbildung 5.2: SyRS - Das Natural Language Modul

- Der **Tokenizer** bereitet den Text für den Tagger auf. Er führt Textersetzungen von Sonderzeichen und Abkürzungen durch. Somit wird der Fehlinterpretation von Satzzeichen entgegengewirkt.

- Der **Tagger** weist jedem Wort aufgrund bestimmter Regeln eine Wortkategorie zu. Diese Informationen werden zur Bildung der Dokumentrepräsentation herangezogen um semantisch bedeutsames Material herauszufiltern.

- Der **Stemmer** wird direkt nach dem Tagger eingebunden. Er führt alle im Text vorhandenen Wörter auf ihre Stammform zurück. Dadurch erhöht sich durch die Reduktion des Wortschatzes sowohl die Performanz des Systems als auch die Ausdrucksstärke der ermittelten Terme (durch Rückführungen grammatikalischer Variationen auf dasselbe Konzept/dieselbe Stammform).

- Die **Stopwortliste** dient der zusätzlichen Reduktion des Wortschatzes. Hier können gängige fachspezifische Begrifflichkeiten, die als allgemein bekannt vorausgesetzt werden können, von der Bildung der Repräsentation ausgeschlossen werden.

- Bei der **Gewichtung** werden die Texte auf Satzebene analysiert und bedeutungstragende Einheiten aufgrund der Taggerinformation identifiziert. Dies geschieht unter Berücksichtigung der syntaktisch korrekten Wortstellung des Deutschen, auf die der Tagger aufbaut. Die so gefundenen Terme werden aus dem Dokument extrahiert und deren Beitrag zur Bildung der Dokumentrepräsentation berechnet. Das Ergebnis stellt ein gewichteter Vektor aller identifizierten Indexterme dar.

Durch die große Anzahl verschiedenster Positionierungsmöglichkeiten von Satzelementen innerhalb von Sätzen ist die Syntax des Deutschen, im Gegensatz zum Englischen, schwer zu fassen [90, 57]. Deshalb wurden zwei verschiedene Versionen von SyRS entwickelt. Diese unterscheiden sich in Bezug auf die verwendeten Termgewichtungsverfahren.

In einer ersten Version wird das Standard-Vektorenmodell verwendet, um den semantischen Inhalt eines Dokuments zu repräsentieren. Das Ergebnis der Analyse jedes Dokuments ist ein gewichteter Wortvektor, der alle bedeutungstragenden Wörter enthält. Die Gewichtung selbst erfolgt über die Termfrequenz sowie über die inverse Dokumentfrequenz (siehe Kapitel 4.3). Alle identifizierten Terme sind bedeutungsmäßig gleichgestellt.

In einer zweiten Version hingegen kommt eine Variante der Thema-Rhema Theorie gemeinsam mit einem Rhema-Thema-Mapping zum Einsatz. Hierbei findet zusätzlich zur Termextraktion eine Unterscheidung von thematischen und rhematischen Termen statt, wobei das Ergebnis in einer sogenannten Thema-Rhema Matrix resultiert. Die Gewichtung erfolgt ebenfalls über die Termfrequenz. Es wird jedoch auch der Beitrag des rhematischen Materials zur Bildung des thematischen Inhalts berechnet (siehe Kapitel 4.4).

Das Ergebnis der Analyse der ersten Version ist ein Vektor, der alle bedeutungstragenden Wörter enthält. Alle Terme werden als semantisch gleichwertig betrachtet, worauf auf deren Reihung oder Kontext keine Rücksicht genommen wird. Die zweite Version hingegen differenziert zwischen thematischen und rhematischen Material, wobei das Ergebnis eine sogenannte Thema-Rhema Matrix ist. Basierend auf dieser Matrix wird in einem nächsten Schritt (siehe Kapitel 5.4.2) ein neuer Gewichtsvektor, der nur thematische Terme enthält, berechnet. Diese Variante nimmt dadurch sowohl Rücksicht auf die Reihung als auch den Kontext von Wörtern [6].

Darüber hinaus wird in beiden Versionen die linguistische Analyse auf zwei verschiedenen Ebenen durchgeführt, da der Begriff „bedeutungstragender Wörter“ nicht genau gefaßt werden kann. In einer „light“ Version zieht lediglich Nomen und Verben als bedeutungstragende Einheiten heran. In einer „heavy“ Version fließen zusätzlich zu den Nomen und Verben noch Adjektive und Adverbien ein. Die Testergebnis-

se beider Versionen (und der jeweiligen „light“ und „heavy“ Varianten) werden im nächsten Kapitel dargelegt.

### 5.2.2 Der Tokenizer

Die unformatierten Textdokumente müssen für die weitere Verarbeitung entsprechend aufbereitet werden. Diese Aufgabe übernimmt der auf lexikalischer Ebene arbeitende Tokenizer.

In einem ersten Schritt werden vordefinierte Textersetzungen im Text durchgeführt. Diese dienen der Substituti von nicht zu verarbeitenden Sonderzeichen. Im nächsten Schritt werden Abkürzungen ('*z.B.*', '*usw.*', '*etc.*', '*Hr.*', '*Dr.*', ...) durch ihre ausgeschriebenen Pendants ersetzt. Etwaige andere notwendig erscheinende Texttransformationen, wie etwa eine Auflösung von Bindestrichen ('*4-tägig*', '*rot-weiss-rot*', ...), könnten ebenfalls in diesem Schritt durch entsprechende Regeleinträge in der Ersetzungsliste durchgeführt werden. Dies wurde aber in dieser Arbeit nicht weiter verfolgt. Im Anschluss daran wird der Text für die nächste Komponente, den Tagger, vorformatiert. Alle Wörter und Satzzeichen werden mit einem Leerzeichen voneinander getrennt und jeder Satz wird in eine separate Zeile geschrieben, um dem Tagger als idealer Input zu dienen. Das Satzende stellt eines der Zeichen '.', '?' oder '!' dar.

Ein Textbeispiel dafür könnte folgendermaßen aussehen:

| | |
|---|---|
| (1) | `Hr. Huber hatte gestern seine Führerscheinprüfung. Heute schon kauft er ein (sehr billiges) rotes, schnelles Auto.` |
| | ↓ |
| (2) | `Herr Huber hatte gestern seine Führerscheinprüfung .`<br>`Heute schon kauft er ein ( sehr billiges ) rotes , schnelles Auto .` |

Der Tokenizer selbst ist in Perl implementiert. Das zu verarbeitende Dokument wird in Form eines Arguments übergeben, wobei mehrere Dateien gleichzeitig als Input zulässig sind. Beim Aufruf des Programms wird der gesamte Text des Dokuments in eine Variable eingelesen. Vorkommende Zeilenumbrüche werden dabei ignoriert.

Danach lädt das Programm die Datei mit den vordefinierten Ersetzungsregeln (replace.lst), die zum Zweck der schnellen und einfachen Erweiterung des Tokenizers eingebunden wird. Die Einträge in dieser Datei haben die Form '*pattern=new pattern*' und werden der Reihenfolge entsprechend abgearbeitet.

Das Ergebnis ist eine formatierte Textdatei, auf die die entsprechenden Ersetzungsregeln angewandt wurden. Jeder Satz entspricht einer einzelnen Textzeile. Alle Wörter, Punktuationen (',', '.', ';', '!', '?', ...) und Sonderzeichen ('(', ')', '[', ']', ...) sind durch Leerzeichen voneinander getrennt. Nach der Aufbereitung des Textdokuments durch den Tokenizer wird es weiter an die nächste Komponente, den Tagger, gereicht.

### 5.2.3 Der Tagger

Der Tagger fügt den einzelnen Wörtern syntaktische Informationen in Form von linguistischen Wortkategorien an. Dies geschieht im Text durch ein Anhängen eines Schrägstrichs und eines Kategoriekürzels am Ende jedes Worts, z.B.: Haus/NN, laufen/VINF, groß/ADJA,...

Syntaktisch bedeutende Informationen können in grammatikalisch richtig geschriebenen Sätzen aufrund ihrer Positionierung und ihres Kontextes identifiziert werden. Um syntaktische Merkmale dem Text hinzuzufügen, verwendet SyRS einen Freeware Tagger namens *tree-Tagger*, der im Netz [1] frei erhältlich ist.

Der auf Satzebene arbeitende Tagger weist jedem Wort eine vordefinierte Wortkategorie zu. Die Funktionsweise des *tree-Taggers* basiert auf der Wahrscheinlichkeitsbestimmung einer Wortkategorie für ein Wort. Dazu wird ein binärer Entscheidungsbaum zusammen mit einer Dreierkette von Wörtern, einem sogenannten Trigramm, verwendet (siehe Abbildung 5.3). Die Wahrscheinlichkeit eines gegebenen Trigramms wird bestimmt, indem der Baum von der Wurzel bis zu einem Blattknoten durchwandert wird. Sucht man beispielsweise nach der Wahrsheinlichkeit eines Nomens (NN), dem ein Artikel (DET) und ein Adjektiv (ADJA) vorgestellt sind, so kann die Wahrscheinlichkeit ausgedrückt werden als $p(NN|DET, ADJ)$. Man beginnt den Test an der Wurzel. Da das Vorgängertag $ADJA$ war, wird entlang der '*ja*' Kante gegangen.

[1] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger-de.html (Stand: 2002.10.17)

Der nächste Test bezieht sich wiederum auf den Vorgänger von '*ADJA*', der '*DET*' ist. Wiederum wird der Kante '*ja*' entlangegangen, über die man einen Blattknoten erreicht. Hier wird nun diejenige Kategorie mit der größten Wahrscheinlichkeit vergeben.

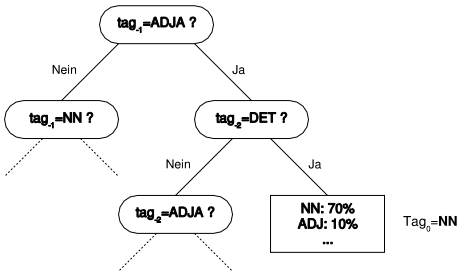



Abbildung 5.3: Entscheidungsbaum, aus [78, Seite 3]

Der Entscheidungsbaum selbst wird rekursiv aus einer Menge von Trainingsbeispielen aufgebaut. Genaueres dazu kann man bei Salton [78, 79] finden.

Die vom Tagger verwendeten Kategorien folgen dem Stuttgart-Tübigen Tagset (STTS). Dieses POS-Tagset (Part-of-Speech-Tagset) wurde 1995 an der Universität Tübingen entwickelt. Dabei werden 11 Hauptwortarten unterschieden: Nomen (N), Verben (V), Artikel (ART), Adjektive (ADJ), Pronomen (P), Kardinalzahlen (CARD), Adverbien (ADV), Konjunktionen (KO), Appositionen (AP), Interjektionen (ITJ) und Partikel (PTK). Weitere Informationen über den *tree-Tagger* und das POS-Tagset finden sich in [3, 78, 79]. Eine Liste der in SyRS verwendeten Tags ist in Tabelle 5.1 angeführt.

Um den oben genannten Entscheidungsbaum oder aber auch andere Regeln (sowohl auf Wort- als auch auf Satzebene) zu finden, muss der Tagger speziell für eine bestimmte Sprache trainiert werden. Dies wird durch ein Training mit großen Datenmengen erreicht, wobei der Lernprozess im eigentlichen Sinn nie abgeschlossen ist. Es bedarf einiges an Zeit- und Arbeitsaufwand, um einen Tagger selbst zu trainieren, weshalb bei SyRS auf einen bereits trainierten Tagger zurückgegriffen wurde.

Ein Tagger liefert im Regelfall kein hundertprozentig richtiges Ergebnis, wie es etwa beim Einsatz von Natural Language Syntaxparsern möglich ist, die sowohl syntaktische als auch semantische Informationen in die Analyse mit einbeziehen. Dem entgegen steht die Performanz solcher Parser beim real-time Einsatz in IR Systemen.

Tabelle 5.1: Verwendetes POS-Tagset, aus [3, Seite 8]

| **POS-Tag** | **Beschreibung** | **Beispiele** |
|---|---|---|
| ADJA | attributives Adjektiv | das [große] Haus |
| ADJD | adverbiales oder prädikatives Adjektiv | er fährt [schnell] |
| | | er ist [schnell] |
| ADV | Adverb | schon, bald, doch |
| NN | normales Nomen | Tisch, Herr, das [Reisen] |
| NE | Eigennamen | Hans, Hamburg, HSV |
| VVFIN | finites Verb, voll | du [gehts], wir [kommen] an |
| VVINF | Infinitiv, voll | gehen, ankommen |
| VVIZU | Infinitiv mit 'zu', voll | anzukommen, loszulassen |
| VVPP | Partizip Perfekt, voll | gegangen, angekommen |
| VAFIN | finites Verb, aux | du [bist], wir [werden] |
| VAINF | Infinitiv, aux | werden, sein |
| VAPP | Partizip Perfekt, aux | gewesen |
| VMFIN | finites Verb, modal | dürfen, können, wollen |
| VMINF | Infinitiv, modal | wollen |
| VMPP | Partizip Perfekt, modal | er hat [gekonnt] |
| PDAT | attribuierendes Demonstrativpronomen | [jener] Mensch |
| PIS | substituierendes Indefinitpronomen | keiner, viele, man, niemand |
| PPER | irreflexives Personalpronomen | ich, er, ihm, mich, dir |
| PPOSS | substituierendes Possessivpronomen | meins, deiner |
| PRELS | substituierendes Relativpronomen | der Hund, [der] |
| PRELAT | attribuierendes Relativpronomen | [mein] Buch, [deine] Mutter |
| PRF | reflexives Personalpronomen | sich, einander, dich, mir |
| PWS | substituierendes Interrogativpronomen | wer, was |

Meist sind die Wartezeiten unakzeptabel, weshalb der Einsatz sehr viel schnellerer Tagger auf diesen Gebieten überwiegt. Da auch SyRS als real-time System entwickelt wurde, kam auch hier aus Performanzgründen ein Tagger zum Einsatz.

### 5.2.4 Der Stemmer

Zur Rückführung eines Wortes auf dessen Stammform werden sogenannte Stemmer verwendet. Diese arbeiten auf Wortebene, wobei der Vorgang des Stemmens von einzelnen Stringmanipulationsregeln gesteuert wird. Diese Regeln entsprechen den verschiedenen Wortbildungsregeln für das Deutsche. Ziel ist es, die Stammform eines beliebigen Wortes zu erzeugen.

Der Vorteil eines effektiven Stemmens liegt vorwiegend in der Reduktion des Vokabulars. Zum Beispiel können die Wörter '*Buch-*', '*Buch-es*' und '*Büch-er*' auf die Stammform '*Buch*' zurückgeführt werden. Auf diese Weise werden verschieden geschriebene Wörter mit gleicher Stammform auf ein und dieselbe Zeichenkette reduziert. Besondere Auswirkungen hat dies bei oft auftretenden Nomen, Verben und Adjektiven, bei denen die Stammform dieselbe ist. Die Performanz des Systems verbessert sich deutlich aufgrund der stark reduzierten Zahl von Worteinträgen.

Ein weiterer Vorteil dieses Zusammenfassens liegt in der Bildung ausdrucksstärkerer Wortreihen, da stammformgleiche Wörter unterschiedlicher Wortkategorien nicht mehr als einzelne Einträge mit einer Kardinalität von 1, sondern unter einem Begriff mit einer Kardinalität von $n$ zusammengefasst werden können. Dies kann als eine semantische Zusammenführung verschiedener Wortformen interpretiert werden. Ein Beispiel dafür ist der Zusammenschluss von '*Wohn-ung*', '*wohn-en*' und '*wohn-lich*' (3 Begriffe mit einer Kardinalität von jeweils 1) zum Begriff '*wohn*' (1 Begriff mit einer Kardinalität von 3).

Im SyRS System kommt ein Freeware Stemmer zum Einsatz der frei im Web[2] erhältlich ist. Dieser Stemmer ist in SnowBall implementiert, einer einfachen und ausdrucksstarken Sprache zur Stringmanipulation. Ein selbstständiges Erweitern oder Ändern des Stemmers ist somit gewährleistet, sollte dies notwendig sein. Außerdem existieren bereits freie Versionen für etliche andere Stemmer, unter anderem für die Sprachen Holländisch, Portugiesisch, Französisch, Spanisch, Englisch, Italienisch, Norwegisch und Schwedisch. Ein mitgelieferter SnowBall-Compiler übersetzt die in Snowball implementierte Grammatik anschließend in performanten native-C Code, der beliebig in andere Anwendungen als simpler Funktionsaufruf, dem ein Ar-

[2] http://snowball.tartarus.org/german/stemmer.html (15.10.2002)

gument übergeben wird, eingebunden werden kann. Der Stemmer selbst wird vor der Termextraktion und der Termgewichtung eingesetzt, um alle im Text vorhandenen Wörter auf ihre Stammformen zu reduzieren.

### 5.2.5 Die Stopwortliste

Zusätzlich werden die Wörter vor der Extraktion noch durch sogenannte Stopwortlisten gefiltert, weshalb diese auch oft als Negativ-Lexikon bezeichnet werden. Vorkommende Wörter werden mittels Patternmatching mit allen in der Liste vorhandenen Wörter verglichen. Entspricht ein Wort einem Wort auf der Liste, wird es als Indexterm verworfen. Auf diese Art kann ein domänenspezifischer Wortschatz, der nicht maßgeblich an der Bedeutungsbildung eines Textes beteiligt ist, ausgefiltert werden. Stopwortlisten werden in der Regel manuell von einem Experten angelegt und erweitert. Es existieren aber auch Methoden zur automatischen Generierung von Stopwortlisten. Beispielsweise können die am häufigsten und die am wenigsten häufig auftretenden Wörter von Texten auf diese Liste gesetzt werden. Fraglich ist allerdings, ob dieses Vorgehen wirklich zu angebrachte Wortlisten führt.

Da Stopwortlisten domänenspezifisch anzulegen sind und das hier verwendete Corpus aus Zusammenfassungen von Diplomarbeiten und Dissertationen aus verschiedensten Bereichen besteht (siehe dazu Kapitel 6.3), wird dieses Konzept in SyRS nur spärlich benutzt. Oft vorhandene Wörter, die hierbei ausgefiltert werden, sind z.B. '*Kapitel*' oder '*Zusammenfassung*'. Da die Stopwortliste erst nach dem Stemming zum Einsatz kommt, sind die Einträge ebenfalls in Form von Wortstammformen anzugeben.

## 5.3 Die Gewichtung

### 5.3.1 Aufbau

Bei der Gewichtung werden bedeutungstragenden Wörter aus einem Text extrahiert und ihr Beitrag zur Bildung des Gesamtinhalts bewertet. Als Input dienen die bereits vorverarbeiteten Daten. Ein Dokument wird dazu satzweise eingelesen. Jedes

Wort im Satz wird als ein Tupel der Form (*Stammform*, *Tag*) angesehen, wobei nur bestimmte Wortkategorien (über das Tag) als semantisch bedeutend betrachtet und verwendet werden: Nomen (NN, NE), Verben (VVFIN, VVIMP, VVINF, VVIZU, VVPP, VAPP, VMPP), Adjektive (ADJA) und Adverbien (ADJD).

SyRS verwendet zwei verschiedene Versionen, die sich in der Identifikation von bedeutungstragenden Einheiten unterscheiden. Eine erste Version verwendet das in Kapitel 4.3 beschriebene Vektorenmodell. Dabei wird jedes Wort als gleich ausdrucksstark behandelt, wobei eine Kombination aus der Wortfrequenz, also der Auftrittshäufigkeit, und der inversen Dokumentfrequenz als zentrales Maß zur Gewichtung herangezogen wird.

Die zweite Version bedient sich einer abgeänderten Variante der Thema-Rhema Theorie. Zuerst werden die jeweiligen Themata und Rhemata des Textes identifiziert und in einer Beziehungsmatrix festgehalten. Anschließend werden daraus zwei Kenngrößen, das Topicality Power und das Explanatory Power, berechnet und das Ergebnis in einer separaten Datei gespeichert. Beide Kenngrößen sind Vektoren, wobei das Topicality Power Auskunft über die Anzahl erklärender Rhemata für ein Thema gibt, und das Explanatory Power angibt, wie oft ein Rhema zur Erläuterung von Themata herangezogen wurde.

Zusätzlich wird in beiden Versionen eine Protokolldatei für jedes analysierte Dokument angelegt. Diese Datei enthält das Ergebnis der Informationsextraktion während der linguistischen Analyse und ermöglicht eine anschließende Evaluation des verwendeten Algorithmus. Die Einträge dieses Logfiles entsprechen der Form '*Satznummer* | *Status* | *Wörter* | *Satz*', bzw. '*Satznummer* | *Status* | *Themata/Rhemata* | *Satz*', wobei das Zeichen '|' den Delimeter darstellt.

Für beide Versionen der Gewichtung wurden zwei Varianten ausgearbeitet. Eine „light“ und eine „heavy“ Variante, wobei sich diese lediglich auf verschiedene Wortkategorien beziehen. Die „light“ Variante betrachtet lediglich Nomen und Verben als sinntragende Einheiten. Die „heavy“ Variante bezieht ebenfalls noch Adjektive und Adverbien in die linguistische Analyse mit ein. Die Ergebnisse zu beiden Doppelvarianten befinden sich im Kapitel 6.

Diese Analyseergebnisse der einzelnen Textdokumente dienen danach dem Neuro-

Tabelle 5.2: „light“ und „heavy“ Variante

| **Variante** | **Wortkategorien** | | | |
|---|---|---|---|---|
| | Nomen | Verben | Adjektive | Adverbien |
| „light“ | NN, NE | VVFIN, VVINF, VVIZU, VVPP VAFIN, VAINF, VAPP, VMFIN VMINF, VMPP | | |
| „heavy“ | NN, NE | VVFIN, VVINF, VVIZU, VVPP VAFIN, VAINF, VAPP, VMFIN VMINF, VMPP | ADJA | ADV, ADJD |

nalen Netzwerk Modul als Input.

### 5.3.2 Das Vektorenmodell

Das Vektorenmodell wird zur Evaluation des Thema-Rhema Modells als Referenzmodell eingesetzt. Deshalb werden hier sowohl dieselben Schritte der Textaufbereitung (Tokenizing, Tagging, Stemming und Stopwortliste) als auch dieselben syntaktischen Wortkategorien des Taggers berücksichtigt. Es existiert ebenfalls eine „light“ und eine „heavy“ Variante, wie bereits in vorhergegangenen Kapitel beschrieben wurde.

### 5.3.3 Das Thema-Rhema Modell

Dieses Modell basiert auf der Thema-Rhema Theorie, die von einem roten Faden ausgeht, der sich durch das ganze Dokument zieht. Diesem Faden entlang werden bestimmte textuelle Inhalte aufgebaut und miteinander verknüpft, welche in die syntaktische Struktur einer Sprache eingewoben werden. Wichtige Informationen sind naturgemäß am Anfang eines Satzes zu finden. Man bezeichnet solche Elemente als *Thema*. Nachdem der zentrale Inhalt eines Satzes (das *Thema*) angeführt ist, werden Ergänzungen, Ausführungen, nähere Eigenschaften und Zusatzinformationen, die dieses Thema näher beschreiben, angeführt. Diese zusätzlichen Informa-

tionen werden als *Rhema* bezeichnet, die ein Thema näher bestimmen. Mit Hilfe dieses Thema-Rhema Konzepts kann nun ein durchgängiger, selbsterklärender Text beschrieben und erfasst werden.

Um die Thema-Rhema Matrix eines Textes zu erstellen, werden die Sätze hintereinander abgearbeitet. Das Finden thematischen und rhematischen Materials innerhalb eines Satzes geschieht anhand des folgenden Algorithmus (siehe Abbildung 5.4):

1. Pro Satz gibt es ein Hauptthema (erstes Nomen im Satz)
2. Pro Satz gibt es beliebig viele Rhemata zum Hauptthema (andere Nomen, Verben, Adjektive, Adverbien)
3. Genauere Analyse auf Nominalphrasenebene (mehrere Themata pro Satz)
   (a) jedes Adjektiv bildet mit seinem nachfolgenden Nomen eine Thema-Rhema Konstruktion

   ```
   das schöne Haus, das neue Auto, ...
   ```

   (b) jedes Adverb bildet mit dem (Voll-)Verb eine Thema-Rhema Konstruktion

   ```
   läuft schnell, springt hoch, ...
   ```

   (c) jedes Nomen bildet mit einem darauffolgenden Nomen eine Thema-Rhema Konstruktion, wenn das Rhema-Nomen in einer direkt auf das Thema-Nomen folgenden Präpositionalphrase vorkommt. Diese Funktionalität ist derzeit noch nicht implementiert.

   ```
   das Auto auf der Wiese, der Garten des Mannes, ...
   ```

Die Thema-Rhema Konzeptbildung auf einer Subebene, wie sie der Schritt 2 in Abbildung 5.4 beschreibt, wird lediglich in der „heavy“ Variante verwendet. Der Versuch soll zeigen, ob eine zusätzliche Berücksichtigung der Nominalphrasenstruktur bei der Bildung der Dokumentrepräsentation erfolgreich durchgeführt werden kann. Die letzte Teilaufgabe des 2. Schrittes (2c) ist derzeit noch nicht implementiert und muss noch rekursiv gelöst werden. Ein Beispiel dazu könnte wie folgt aussehen:

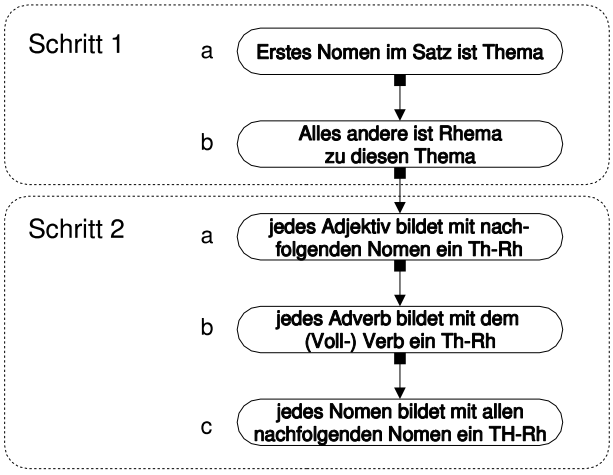



Abbildung 5.4: Die Thema-Rhema Analyse

| | Der Mann | mit dem Auto | der Frau | trägt | heute | einen schönen | Hut. |
|---|---|---|---|---|---|---|---|
| | $Th_1$ | $Rh_1$ | $Rh_2$ | $Rh_3$ | $Rh_4$ | $RH_5$ | $RH_6$ |
| (1a+1b) | $(Th_1, Rh_1 - Rh_2 - Rh_3 - Rh_4)$ | | | | | | |
| (2a) | $(Th_2 = Rh_6, Rh_5)$ | | | | | | |
| (2b) | $(Th_3 = Rh_3, Rh_4)$ | | | | | | |
| (2c) | $(Th_4 = Th_1, Rh_1 - Rh_2)$ | | | | | | |
| (2c) | $(Th_5 = Rh_1, Rh_2)$ | | | | | | |

Im Gegensatz zur ursprünglichen Thema-Rhema Theorie (für das Englische), nach der das Verb die Trennposition zwischen thematischen und rhematischen Material darstellt, wird hier (für das Deutsche) das erste im Satz vorkommende Nomen als das Hauptthema und somit als Trennposition interpretiert. Der Vorteil dieser Annahme liegt in der Möglichkeit, Sätze zu verarbeiten, bei denen das erste vorkommende Nomen nicht notwendigerweise vor dem ersten vorkommenden Verb stehen muss. Da Sätze ohne ein Thema nicht verarbeitet werden, könnten laut der ursprünglichen Theorie Sätze wie etwa

| Heute hat | Klaus | ein neues Auto gekauft. |
|---|---|---|
| | $Thema$ | |

nicht verarbeitet werden. Da solche Sätze im Deutschen jedoch sehr oft vorkommen, wurde durch diesen Schritt dieses Problem umgangen.

Im Gegensatz zur originalen Thema-Rhema Theorie werden ebenfalls nicht alle vor der ersten Verbposition vorkommenden Nomen als Themata angenommen. Dadurch würde das zweite Nomen vor dem ersten Verb als Rhema identifiziert, welches der Theorie nach (im Deutschen) als das klassische Rhema anzunehmen ist:

| `Der Präsident von Israel besucht Palästina.` |
|---|
| *Thema* *Rhema* |

Sollte ein Satz kein Hauptthema aufweisen, wird das in der Protokolldatei speziell vermerkt. Da ein fehlendes Thema bei einer Eintragung in die Thema-Rhema Matrix verworfen wird, fließen diese Sätze nicht in die weitere Analyse mit ein.

Im Gegensatz zum ursprünglichen Prototyp von SyRS für das Englische wurde auch auf eine pronominale Ersetzung verzichtet. Da im Deutschen eine viel freizügigere Positionierung der Elemente innerhalb von Sätzen möglich ist, benötigt man zur Auflösung ein syntaktisch-semantisches Ersetzungssystem. Da jedoch ein Tagger keine Aussage über syntaktische Eigenschaften von Wörtern (Fall, Geschlecht, Zahl, Zeitstufe, ...) treffen kann und diese Informationen auch nicht aus einem Lexikon bezogen werden können (da sie kontextbezogen sind), ist eine solche Ersetzung problematisch. Trotzdem wird in der Protokolldatei festgehalten, das eine pronominale Ersetzung durchzuführen wäre. Dies geschieht durch die vom Tagger vergebenen Tags für Pronomen (PDAT, PIS, PPER, PPOSS, PRELS, PRELAT, PRF und PWS). Für eine genauere Erläuterung der Tags sei hier auf Tabelle 5.1 verwiesen.

Die schematische Darstellung in Abbildung 5.5 (siehe auch Tabelle 4.2) zeigt die zeilen- und spaltenweise Bildung der zwei Hauptgrößen Topicality und Explanatory Power. Es handelt sich hierbei um numerische Häufigkeitswerte für jedes Thema und Rhema. Diese Werte dienen dem nachfolgenden Neuronalen Netzen als Input.

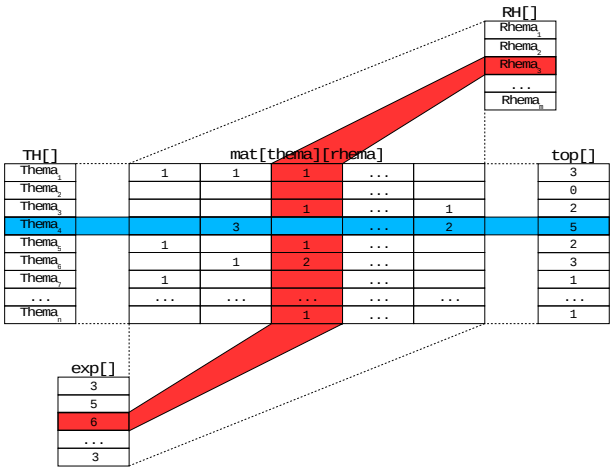



Abbildung 5.5: Version 2: Thema-Rhema Matrix

## 5.4 Das Neuronale Netzwerk Modul

Das Neuronale Netzwerk Modul dient zwei verschiedenen Aufgaben, weshalb auch zwei unterschiedliche Netzwerktypen verwendet werden: ein Fuzzy Associative Memory Netz (FAM) und ein fuzzy Adaptive Resonance Theory Netz (fuzzy ART). Das erstes Netz (FAM) übernimmt eine Rhema-Thema Zuordnung, während das zweite Netz (fuzzy ART) anschließend dazu verwendet wird, das eigentliche Clustering durchzuführen. Beide Netzwerktypen werden in diesem Kapitel ausführlich beschrieben.

### 5.4.1 Aufbau

Das Neuronale Netzwerk Modul besteht aus zwei hintereinandergeshalteten Neuronalen Netzen, mit deren Hilfe ein Dokumentclustering durchgeführt wird (siehe Abbildung 5.6).

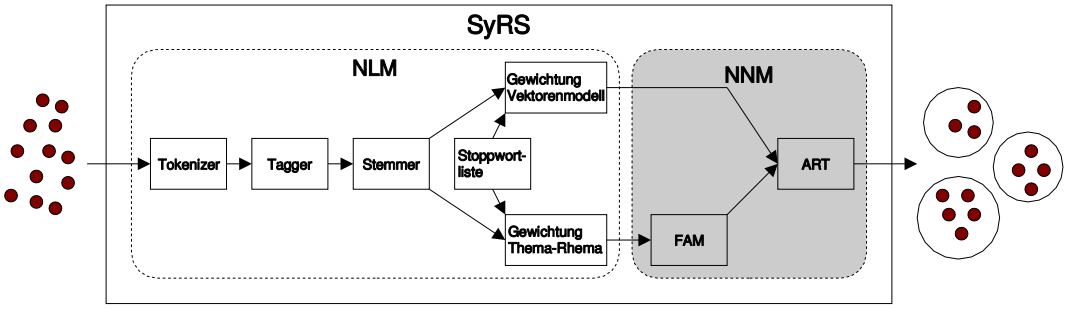



Abbildung 5.6: SyRS - Das Neuronale Netzwerk Modul

Es besteht aus folgenden Komponenten:

- Das Fuzzy Associative Memory (FAM) Netz versucht Ähnlichkeiten zwischen Relationen von rhematischen und thematischen Material festzustellen. Hierbei werden bestimmte Rhemata auf einzelne Themata abgebildet. Dies soll eine Generalisierung von Konzepten (Stammformen) zu Metakonzepten (Inhalten) darstellen. Dieses Netz wird nur im Zusammenhang mit der zweitern Version von SyRS benötigt, dessen Analyse auf der Thema-Rhema Theorie beruht. Es berechnet dabei den Beitrag des rhematischen Materials bei der Gewichtung der thematischen Inhalte. Version 1 von SyRS, das Vektorenmodell, verwendet das FAM Netz nicht, da hier Beziehungen zwischen Wörtern nicht berücksichtigt werden, weshalb auch keine Term-Term Zuordnung durchzuführen ist.

- Das fuzzy Adaptive Resonance Theory (ART) Netz weist jedem Text aufgrund seiner analysierten Daten einen bestimmten Dokumentcluster zu. Die Zuweisungen werden dabei über einen reellen Zugehörigkeitsfaktor gewichtet.

Im Betrieb von SyRS unterscheidet man zwei Modi. Zu Beginn wird das System mit einer entsprechend großen und repräsentativen Menge an Textdokumenten trainiert. Hierbei werden auftretende Ähnlichkeitsmuster im textuellen Inhalt automatisch erkannt und erlernt. Aufgrund dieser gefundenen Ähnlichkeitsmuster von Textdokumenten untereinander werden während des Trainingsmodus sogenannte Dokumentcluster gebildet. Im nachfolgenden Betrieb wird das System im Testmodus gestartet. Aufgrund einer Abfrage werden alle in Frage kommenden Dokumentcluster ermittelt. Als Ergebnis werden alle Dokumente aller passenden Dokumentcluster zurückgegeben. SyRS fokussiert dabei auf Recall, also dem Zurückgeben möglichst aller in Frage kommenden Dokumente. Im nächsten Schritt trifft der Benutzer manuell eine engere Auswahl darüber, welche der bezogenen Dokumente nun wirklich für diese Abfrage relevant sind. Das System unterstützt den Benutzer also bei der Einschränkung des Suchraumes (siehe Kapitel 2.5).

### 5.4.2 Das Fuzzy Associative Memory (FAM) Netz

In der zweiten Version von SyRS, die eine Textanalyse mittels der Thema-Rhema Theorie vornimmt, ist ein Fuzzy Associative Memory Netzwerk dem eigentlichen Clusteringverfahren vorgeschaltet. Dieses Netz versucht, aus dem im Text vorhandenen rhematischen Material neues, im Text nicht explizit genanntes, thematisches Material abzuleiten. Da eine eindeutige Zuordnung eines Rhemas zu einem Thema meist nicht möglich ist, wird hier auf die bereits angesprochenen Fuzzy-Sets (siehe Anhang A) mit ihren Operationen zurückgegriffen. Mittels einer Ähnlichkeitsfunktion wird hierbei eine Liste aller möglichen impliziten Themata zusammen mit einer Gewichtung errechnet. Abbildung 5.7 zeigt diesen Vorgang schematisch.

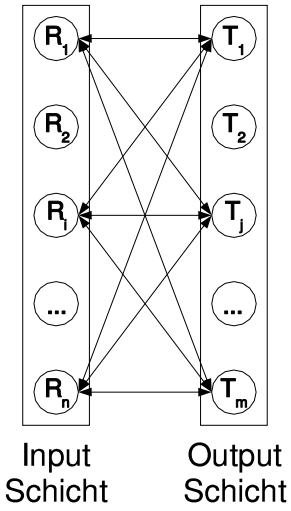



Abbildung 5.7: Das Fuzzy Associative Memory Netz (FAM)

Der Input des Assoziativspeichers besteht aus den vom Natural Language Modul erzeugten Textanalysedokumenten. In diesen Dokumenten befinden sich die Listen der Themata und Rhemata sowie das dazugehörige Beziehungsgitter zwischen ihnen. Ebenfalls sind die Kenngrößen Topicality und Explanatory Power zu den einzelnen Themata und Rhemata bereits berechnet.

Der Inputvektor selbst wird durch die durchschnittlichen Explanatory Powers der Rhemata repräsentiert, die über die Formel

$$\mu_R(R_j) = \frac{\text{Explanatory Power von } R_j}{\text{Summe aller Explanatory Powers}} = \frac{EXP(R_j)}{\sum_{i=0}^{m} EXP(R_i)} \tag{5.1}$$

gegeben ist.

Die Abbildung der Zuordnungen von Rhemata zu Themata spielt die zentrale Rolle beim Speichern innerhalb des Netzes. Dies geschieht in einer Matrix der Form

$$W = R^T T \tag{5.2}$$

wobei $R^T$ dem transponierten Rhemata-Vektor und $T$ dem Vektor der Themata entspricht. Das Erinnern einer Rhemata-Themata Assoziation kann anschließend durch eine Berechnung von

$$T = RW \tag{5.3}$$

durchgeführt werden. Näheres dazu findet sich in [6] ab Seite 132.

Der eigentliche Lernalgorithmus des FAM Netzes ist folgendermaßen gegeben (siehe auch Abbildung 5.8):

1. Initialisierung:
   $\forall j = 1..n, i = 1..m : w_{ij} = 1$
2. Für alle Lernbeispiele $P^{(k)} = [(R_1, .., R_n), (T_1, .., T_m)], k = 1..p$ wiederhole
   (a) Berechne den aktuellen Output:
   $\forall T_i' = \bigvee_{j=1}^{n} (w_{ji} * R_j)$
   (b) Anpassung der Werte:
   $\delta_i = T_i' - T_i$

$$\begin{cases} w_{ji}(t+1) = w_{ji} - \eta\delta i & \text{wenn } w_{ji}(t) * R_j > T_i \\ w_{ji}(t+1) = w_{ji} & \text{sonst} \end{cases}$$

   wobei $\eta$ ein Parameter für die Schrittgröße ist $(0 < \eta \leq 1)$.
   (c) Wiederhole Schritte 2a und 2b bis $w_{ji}(t+1) = w_{ji}(t)\ \forall j, i$

Um dem Effekt des Vergessens von bereits erlernten Wissen von $\eta$ (im Schritt 2) entgegenzuwirken, darf die Maximalanzahl an Lernbeispielen $p^2 < n$ für das Forward

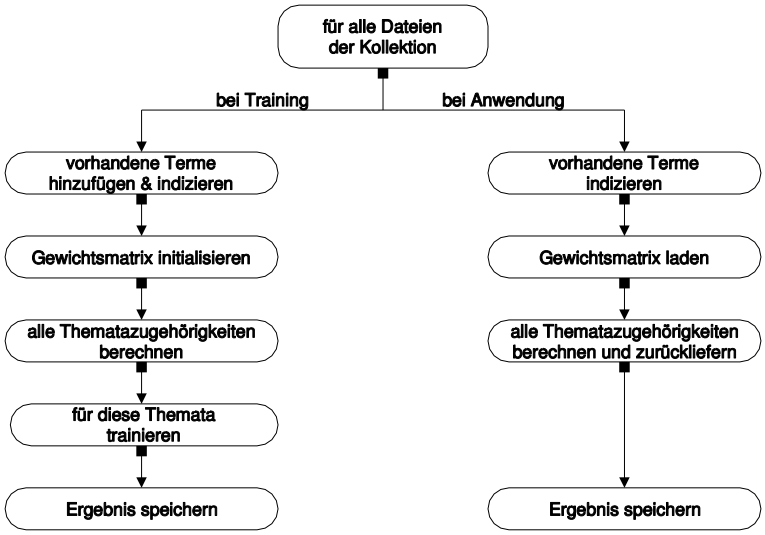



Abbildung 5.8: Mappingalgorithmus des FAM

Learning und $p^2 < m$ für das Backward Learning nicht überschritten werden. Da in SyRS lediglich Forward Learning eingesetzt wird, also das Erlernen von Rhemata zu Themata, muss lediglich die Erste der beiden Bedingungen erfüllt sein.

Das FAM Netz kann über den Parameter $\eta$ gesteuert werden. Diese Parametereinstellung trägt wesentlich zum Verhalten des Netzes bei.

| | |
|---|---|
| **eta** | Der Step Size Parameter $\eta$ gibt die Schrittgröße des FAM an, mit dem sich die Werte der Gewichtsmatrix dem Idealzustand nähern. Ein größeres Intervall entspricht einer schnelleren Konvergenz, allerdings können dadurch lokale Minima auftreten. Dies kann durch eine Beschränkung der Maximalanzahl der Beispieldaten während des Trainings ($\#Samples < min(m, n)$) verhindert werden [6]. |

Das FAM Netz führt somit eine Rhema-Thema-Zuordnung durch und speichert das erlernte Ergebnis in einer Gewichtsmatrix. Auf diese Weise ermittelte (implizite) Themata werden mit den im Text bereits vorhandenen (expliziten) Themata kombiniert (siehe Kapitel 5.4.4) und dienen im nächsten Schritt dem fuzzy ART Neztwerk als Input, welches das eigentliche Clusteringverfahren durchführt.

Das FAM wird in SyRS eingesetzt, um versteckte Muster in Dokumenten zu erkennen. Sobald eine Menge von Rhemata in einem Text auftritt, und diese Rhemata ein bestimmtes Konzept beschreiben, das jedoch nicht explizit im Text genannt ist, so

aktiviert das FAM dieses Konzept automatisch. Dies ist eine andere Form der Handhabung von Unschärfe und ein Versuch, unvollständige oder abstrakte Sachverhalte ebenfalls mit einfließen zu lassen.

### 5.4.3 Das fuzzy Adaptive Resonance Theory (ART) Netz

Das fuzzy Adaptive Resonance Theory Netz bildet die letzte Stufe von SyRS. In diesem Schritt werden die gewichteten Dokumentrepräsentationen bestimmten Dokumentclustern zugeordnet.

Die Stärke von ART Netzen liegt in ihrer Fähigkeit Kategorisierungen vorzunehmen und Muster zu erkennen. Dabei werden zwei wichtige Phenomäne in Bezug auf Neuronale Netze berücksichtigt: Stabilität und Plastizität. Stabilität benennt dabei das Produzieren akzeptabler Ergebnisse in einer sich ändernden Umgebung (Input-Akzeptanz). Plastizität meint das Anpassen des Netzwerkoutputs an seine Umwelt (Output-Akzeptanz). Diese zwei Eigenschaften erlauben es ART Netzen, eine Ausgeglichenheit zwischen bereits erlernten und neuen Mustern beizubehalten.

ART Netze sind selbstlernend und ermöglichen eine Klassifikation von Mustern in Kategorien. Ein fuzzy ART Netz besteht aus drei Schichten (siehe Abbildung 5.9): einer Input-Schicht, einer Vergleichs-Schicht und einer Output- oder Kategorien-Schicht. Die Input-Schicht reicht dabei die angelegten Werte ohne weitere Verarbeitung eins zu eins weiter an die Vergleichs-Schicht. Jede Einheit der Output-Schicht speichert einen Prototyp (ein Clusterzentrum) einer Kategorie (eines Clusters). Jede Einheit der Vergleichs-Schicht ist mit jeder Einheit der Output-Schicht verbunden.

ART unterscheidet nicht zwischen einer Trainings- und einer Testphase, wie es bei anderen, auf maschinellen Lernen basierenden Systemen notwendig ist. Lernen wird hier als eine kontinuierliche Anpassung der Kantengewichte angesehen, die bei jedem neuen Input vollzogen wird. Die Einheiten $j$ der Output-Schicht führen einen Wettbewerb durch, um den angelegten Input $T$ zu repräsentieren. Es wird diejenige Einheit $J$ der Ouptut-Schicht mit dem höchsten Aktivierungslevel ausgewählt. Das Aktivierungslevel der Einheit $J$ kann durch die Formel

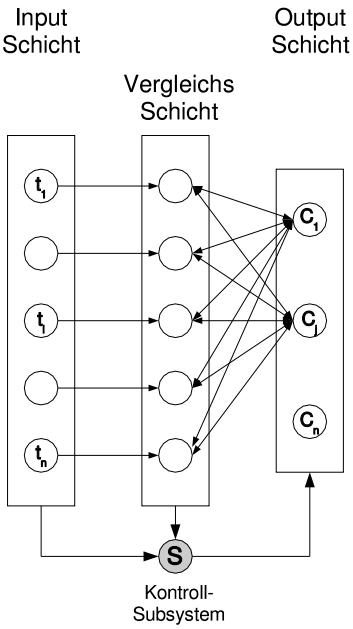



Abbildung 5.9: Das Adaptive Resonance Theory Netz (ART)

$$H_j = \frac{|T \cap W_j|}{\alpha + |W_j|} \tag{5.4}$$

ausgedrückt werden. $W_j$ repräsentiert die Gewichte der Kanten zwischen der Einheit $j$ der Output-Schicht und dem Inputvektor $T$ der Termschicht. $\cap$ entspricht dem fuzzy AND Operator, $|...|$ repräsentiert die fuzzy Kardinalität und $\alpha$ ($> 0$) ist ein benutzerdefinierter Parameter.

Das Aktivierungspotential der Gewinnerkategorie wird über die gewichteten Kanten zur Vergleichsschicht backpropagiert. Das Netz berechnet einen Grad an Ähnlichkeit zwischen dem Prototyp und dem Input in der Form

$$\frac{|T \cap W_J|}{|T|} \leq \rho \tag{5.5}$$

Ist die Kategorie dem angelegten Input entsprechend ähnlich genug, erreicht das Netz einen Resonanzzustand. Die Ähnlichkeitsbestimmung erfolgt durch einen einfachen Test zur Bestimmung der Überlappung zwischen dem Input und dem Output und einem Grenzwert $\rho$, dem sogenannten Vigilance Parameter, der angibt wie groß eine Überschneidung sein muss. Je größer dieser Parameter gewählt wird, desto größer muss eine Übereinstimmung sein. Ein größerer Wert für $\rho$ führt also zur Bildung einer größeren Anzahl kleinerer Kategorien.

In einem solchen Fall wird das durch $T$ repräsentierte Dokument der resonanten Kategorie zugewiesen. Die Kantengewichte zwischen der Gewinnerkategorie und dem Input werden entsprechend der Formel

$$W_J^{new} = \beta(T \cap W_J^{old}) + (1 - \beta)W_J^{old} \tag{5.6}$$

angepasst. $\beta(0 < \beta \leq 1)$ wird als Lernparameter bezeichnet. Der Spezialfall von $\beta = 1$ ermöglicht ein schnelles Lernen.

Die Anpassung der Kantengewichte verstärkt die Verbindung zwischen der resonanten Kategorie und dem Inputvektor. Sobald derselbe Input am Netz angelegt wird, ist das Aktivierungslevel der Kategorie höher.

Ist die Übereinstimmung des Inputs mit einem Kategorienprototyp nicht ausreichend (in Bezug auf den Vigilance Parameter $\rho$), so wird durch das Kontroll-Subsystem ein Reset durchgeführt und die Kantengewichte zur Kategorie nicht verändert. Der Vorgang startet nun bei der nächsten Kategorie mit derselben Vergleichsprozedur.

Für den Fall das keine der vorhandenen Kategorien dem Inputvektor ähnlich genug ist, verhält sich ART anders als übliche NN. Es setzt seine Fähigkeit der Anpassung (Plastizität) ein, um mit dem Input eine neue Kategorie anzulegen. In einem solchen Fall wird die Output-Schicht um ein Element vergrößert, die mit allen Einheiten der Vergleichs-Schicht verbunden wird. Die Kantengewichte werden dabei entsprechend mit 1 initialisiert. Alle Kanten von Indextermen, die im Inputvektor auftreten, werden durch die Formel

$$T_{normalisiert} = \frac{T}{|T|} \tag{5.7}$$

normalisiert gewichtet. Diese Normalisierung verhindert ein zu rasches Anwachsen vorhandener Kategorien. Der eigentliche Algorithmus kann wie folgt angegeben werden (siehe auch Abbildung 5.10):

1. Initialisierung:
   $\forall j = 1..n(\ \#\ \text{Kategorien}), i = 1..m : w_{ij}(0) = 1$
   $0 < \rho \leq 1, 0 < \alpha, 0 < \beta \leq 1$ (wenn $\beta = 1$ : schnelles Lernen )

2. Für alle Inputvektoren $T = [T_1, .., T_m]$ wiederhole

   (a) Berechne die Kategorie für den aktuellen Input:
   Für alle Kategorien $j$ berechne $H_j = \frac{|T \cap W_j|}{\alpha + |W_j|}$
   Die Gewinnerkategorie $J$ ist $\max(H_1, .., H_n)$
   Im Falle mehrerer Gewinnerkategorien wähle die Erste

   (b) Resonanzzustand:
   Tritt auf, sobald die Bedingung $\frac{|T \cap W_J|}{|T|} \leq \rho$ erfüllt ist

3. Lernen:
   Wenn ein Resonanzzustand erreicht wurde, Kantengewichte anpassen
   $W_J^{new} = \beta(T \cap W_J^{old}) + (1 - \beta)W_J^{old}$

4. Neue Kategorie anlegen:
   Wenn kein Resonanzzustand erreicht wurde, neue Kategorie mit Inputvektor als Prototyp anlegen

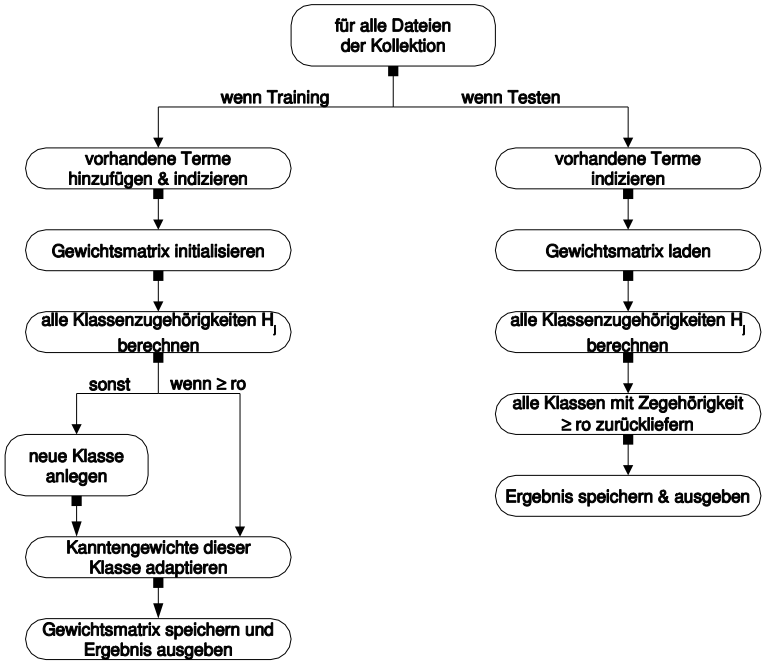



Abbildung 5.10: Clusteralgorithmus des fuzzy ART

Das fuzzy ART führt nicht nur ein auf Prototypen basierendes Clustering durch, sondern erlaubt ebenfalls die Bildung hierarchischer Cluster. Dies wird bewerkstelligt

durch eine Veränderung des Parameters $\rho$, der bei größeren Werten zu mehr Kategorien mit weniger Dokumenten oder bei niedrigeren Werten zu weniger Kategorien mit mehr Dokumenten führt.

Das ART Netz kann über drei Parameter gesteuert werden. Diese Parametereinstellungen tragen wesentlich zum Verhalten des Netzes bei.

| | |
|---|---|
| **alpha** | Der Choice Parameter $\alpha$ dient dazu, den Netzwerkoutput während der Klassenauswahl unter 1 zu halten. Der Standardwert in SyRS ist 0.05. |
| **rho** | Der Vigilance Parameter $\rho$ bestimmt den Grenzwert des Ähnlichkeitsgrades, der überschritten werden muss, damit ein Output als passend erachtet wird. |
| **beta** | Der Learning Parameter $\beta$ legt fest, inwieweit sich die Werte in der Gewichtsmatrix ändern. Ein Wert von 0 bedeutet, dass sich die Gewichtsmatrix nicht ändert. Ein Wert von 1 hingegen bewirkt eine Änderung ohne Rücksichtnahme auf den vorherigen Wert in der Gewichtsmatrix (schnelles Lernen). Die Standardeinstellung in SyRS ist 1.00. |

### 5.4.4 Zusammenführung von FAM und ART

Die zweite Variante von SyRS verwendet das FAM Netz, um zusätzliche Themata in der Dokumentrepräsentation zu bilden. Hierbei wird versucht, den Dokumentinhalt mittels expliziter und impliziter Themata zu charakterisieren. Explizite Themata im Text sind dadurch gekennzeichnet, dass sie im Text genannt sind. Im Gegenzug dazu bezeichen implizite Themata Inhalte des Textes, die selbst nicht im Text vorkommen, jedoch um die sich ein Text dreht.

So könnten beispielsweise die Begriffe '*Speicher*', '*Festplatte*' und '*Tastatur*' auf das explizit im Text genannte Thema '*Computer*' abgebildet werden. Es könnten jedoch auch die Begriffe '*Planze*', '*grün*', '*Baum*', '*Tanne*' und '*Lichtung*' auf ein im Text nicht vorhandenes, implizites Thema, wie etwa '*Wald*', führen.

Es ist nun notwendig, die expliziten mit den impliziten Themata zu vereinen. Dies geschieht über die fuzzy Vereinigung der Form:

$$T' = \mu_{T_{implizit} \cup T_{explizit}} \tag{5.8}$$

Dadurch wird jedem thematischen Term, egal ob explizit oder implizit im Text vorhanden, das maximale Gewicht zugeordnet (siehe Abbildung 5.11). Die Motivation dahinter ist eine Verbesserung des Recalls, da implizite Themata zur Bildung des Gesamtinhaltes beitragen.

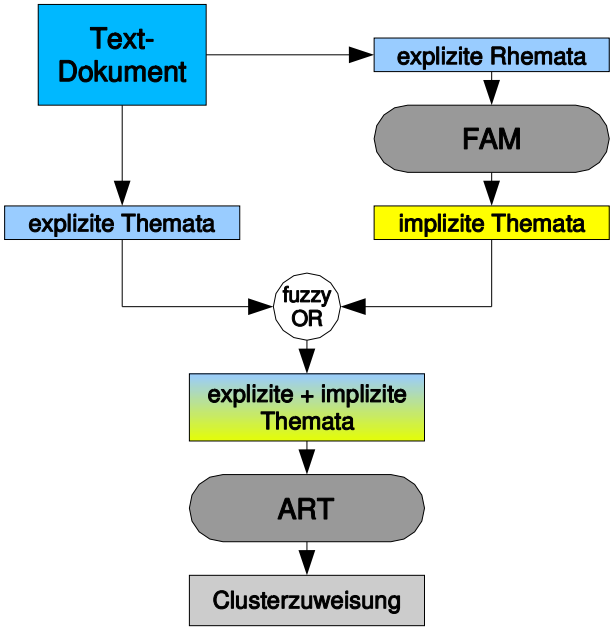



Abbildung 5.11: Rhema-Thema Mapping (FAM) und Clustering (ART)

KAPITEL 6

# Evaluation von SyRS

Dieses Kapitel beschäftigt sich mit der Evaluation von SyRS für deutschsprachige Texte eingegangen. Hierzu werden die Grundvoraussetzungen einer Evaluation zusammen mit verschiedenen Evaluationsmetriken vorgestellt, die eine Auswertung solcher Systeme ermöglichen. Weiters erfolgt die Beschreibung des verwendeten Corpus sowie möglicher Alternativen und im Anschluss daran die Präsentation und der Vergleich der erzielten Ergebnisse. Zusätzlich wird ein Ansatz zur Anwendung von Textclustering für den Bereich des Information Retrievals vorgestellt.

## 6.1 Einleitung

Nach der Vorstellung der Architektur und der Arbeitsweise von SyRS im vorhergegangenen Kapitel, wird in diesem Kapitel eine experimentelle Auswertung vorgenommen. Die Resultate der zwei Gewichtungsverfahren, die zum Einsatz kamen, das Vektorenmodell und die Thema-Rhema Analyse, wurden einzeln ausgewertet und miteinander verglichen. Das Hauptaugenmerk richtet sich darauf, herauszufinden, inwieweit sich die Thema-Rhema Analyse zur Bildung der Dokumentrepräsentation und einer anschließenden Gewichtung deutscher Texte eignet.

Das Vektorenmodell extrahiert bei der Repräsentation des Inhalts von Texten lediglich Indexterme. Beziehungen zwischen Indextermen werden nicht berücksichtigt. Da es oft als Standard-Modell eingesetzt wird, sollte es sich als neutrales Vergleichsmedium eignen. Im Gegensatz zum Vektorenmodell berücksichtigt die Thema-Rhema Analyse Beziehungen zwischen Termen (durch die „wird erklärt durch“ Beziehung). Untersucht werden soll die Aussagekraft dieser Beziehung und das Vorgehen bei der Identifikation solchen Materials. Durch eine Gegenüberstellung mit dem Vektoren-

modell soll eine Abschätzung der Effizienz dieser Analyse vorgenommen werden.

Nach der Beendigung der Entwicklung von IR Systemen muss das System evaluiert werden, um die Performanz zu ermitteln und mit anderen Systemen vergleichbar zu sein [4]. Die Art der Evaluation hängt dabei natürlich vom Ziel des IR Systems ab. Die Funktionalität des Systems steht im Vordergrund, und diese wird Schritt für Schritt getestet. Besteht das System diesen Test, werden Performanztests durchgeführt. Hier sind die zwei Kenngrößen Zeit und Platz entscheidend. Je kürzer die Verarbeitungszeit und je geringer der Speicherverbrauch, desto besser ist ein System. Hierbei handelt es sich um einen immerwährenden Kompromiss zwischen diesen beiden Größen [4, 6].

Bei IR Systemen, die Textgruppierungen vornehmen, sind jedoch auch andere Metriken von Interesse. Da die gebildeten Kategorien bzw. Cluster mit ihren Dokumenten keine exakten Ergebnisse darstellen, muss die automatisch entwickelte Kategorienstruktur einer menschlichen Kategorisierung gegenübergestellt werden. Eine solche Auswertung basiert auf einem vorkategorisierten Dokumentcorpus (Trainings- und Testkollektion) und einer Menge von Evaluationsmetriken. Die Aufgabe einer Evaluationsstrategie ist es, den Übereinstimmungsgrad zwischen einer automatischen und einer menschlichen Kategorisierung eines Experten zu quantifizieren. Dies erlaubt eine Abschätzung der Performanz eines Textgruppierungssystems.

Um eine Textgruppierung zu beurteilen, sind drei Grundvoraussetzungen notwendig [6]:

- Zum einen wird eine geeignete Dokumentkollektion, ein Corpus, benötigt. Die Texte der Kollektion müssen dazu analysekonform sein, d.h. der Parser muss die Texte verarbeiten können. Für den in dieser Arbeit vorliegenden Prototyp ist dieser insofern Punkt entscheidend, da die einzelnen Sätze anhand der Thema-Rhema Analyse analysiert werden. Je mehr Sätze hiervon die Ausnahme bilden, desto mehr wird das Ergebnis der Repräsentation beeinträchtigt. Einzelne Ausnahmen beeinflussen das Ergebnis nur geringfügig und können deshalb vernachlässigt werden, ein zu grosser Anteil führt jedoch zu schlechten Ergebnissen.

- Die einzelnen Texte des Corpus müssen entsprechend vorkategorisiert sein. Dies erlaubt eine Auswertung des erzielten Ergebnisses durch einen Vergleich mit einer (meist von Experten durchgeführten) Idealkategorisierung. Auf das Problem einer menschlichen Fehlkategorisierung soll an dieser Stelle hingewiesen sein. Da jedoch das Ergebnis des Clusterings mit irgendeiner Vorkategorisierung verglichen werden muss, wird diese Problematik im Zuge dieser Arbeit nicht genau erläutert.

- Es werden Evaluationsmetriken benötigt, die eine Bewertung des Systems ermöglichen. In dieser Arbeit werden traditionelle Kennwerte wie Recall, Precision, Accuracy, Error und Fallout verwendet. Im weiteren Verlauf dieses Kapitels wird auf diese Metriken genauer eingegangen.

## 6.2 Evaluationsmetriken

### 6.2.1 Precision und Recall

Die weitverbreitetsten Kennzahlen zur Bewertung von Textgruppierungen basieren auf dem Modell der Kontingenztabelle [72, 51, 82, 81, 93]. Von einem System müssen $n$ Entscheidungen (für jedes Dokument eine) getroffen werden, wovon jede genau eine richtige Antwort hat (Ja oder Nein). Das Ergebnis dieser $n$ Entscheidungen kann in einer Kontingenztabelle (siehe Tabelle 6.1) festgehalten werden.

Tabelle 6.1: Kontingenztabelle (Multiple Binary Classification), aus [81, Seite 42]

| Gruppierung anhand | | Expertenentscheidung | | Summe |
|---|---|---|---|---|
| | | **JA** | **NEIN** | |
| System-entscheidung | **JA** | $a$ | $b$ | $a+b$ |
| | **NEIN** | $c$ | $d$ | $c+d$ |
| Summe | | $a+c$ | $b+d$ | $a+b+c+d=n$ |

Aus dieser Tabelle lassen sich die Kennzahlen Precision (6.1) , Recall (6.2), Error (6.3), Accuracy (6.4), Overlap (6.5) und Fallout (6.6) definieren [51, 40, 96, 93, 1].

$$P = \frac{a}{a+b} = \frac{\text{Anzahl von gefundenen Kategorien die richtig sind}}{\text{Summe aller gefundenen Kategorien}} \tag{6.1}$$

$$R = \frac{a}{a+c} = \frac{\text{Anzahl von gefundenen Kategorien die richtig sind}}{\text{Summe aller (gefundenen und nichtgefundenen) Kategorien}} \tag{6.2}$$

$$E = \frac{b+c}{a+b+c+d} \tag{6.3}$$

$$A = \frac{a+d}{a+b+c+d} \tag{6.4}$$

$$O = \frac{a}{a+b+c} \tag{6.5}$$

$$F = \frac{b}{b+d} \tag{6.6}$$

Äquivalente Kennzahlen zu Recall und Fallout wurden erstmals in der Signalentdeckung (Signal Detection Theory) eingesetzt [84]. Recall gibt das Verhältnis richtig zugewiesener Dokumente zur Anzahl aller richtig zuzuweisenden Dokumenten an, während Precision den Anteil richtiger Zuweisungen zu allen getätigten Zuweisungen angibt. Fallout stellt den Gegenpol von Recall dar, da hier das Verhältnis falsch zugewiesener Dokumente zu allen nicht zuzuweisenden Dokumenten gemessen wird. Overlap ist symmetrisch gegenüber $b$ und $c$ und wird verwendet, um zwei Kategorisierungen miteinander zu vergleichen ohne festzulegen, welche der beiden korrekt ist. Accuracy gibt die Anzahl richtig zugewiesener Dokumente zu allen $n$ getroffenen Entscheidungen an. Error stellt wiederum den Gegenpol von Accuracy dar, da es den Anteil falscher Zuweisungen aller getroffenen Entscheidungen bestimmt.

Obwohl einzelne Metriken für sich gestellt keine guten Indikatoren für die Performanz eines Textgruppierungssystems darstellen, bedeutet dies nicht, dass sie nutzlos sind. Besonders in Kombination mit anderen Kennzahlen erlauben sie einen tieferen Einblick in die Leistungsfähigkeit eines solchen Systems. Bei diesem Experiment kamen die fünf Kennzahlen Recall, Precision, Fallout, Accurancy und Error zum Einsatz.

Micro-Averaging und Macro-Averaging [50, 72, 81, 93, 27, 35, 53, 1] sind zwei gängige Methoden, diese Kennzahlen zu errechnen. Beim Macro-Averaging werden die einzelnen Kennzahlen für jede Kategorie einzeln berechnet. Das endgültige Ergebnis wird durch die Bildung des Durchschnitts der einzelnen Werte bestimmt. Micro-Averaging hingegen berechnet die Werte der Kontingenztabelle ($a$, $b$, $c$ und $d$) global

für alle Zuweisungen. Dadurch wird jede Kennzahl genau einmal für eine gesamte Kollektion berechnet.

Diese zwei Vorgehensweisen liefern unterschiedliche Ergebnisse. Macro-Averaging behandelt alle Kategorien als gleichgewichtet und bildet somit einen Durchschnitt pro Kategorie. Micro-Averaging hingegen erlaubt es, einzelne Dokumente gleich zu gewichten und bildet somit einen Durchschnitt pro Dokument (einen Durchschnitt über alle Dokument/Kategorie - Paare) [1, 93].

In der Literatur herrscht keine Einigkeit darüber, welches dieser Verfahren sich besser für die Evaluation von Textgruppierungen eignet [81, 6]. Einige sind der Meinung, dass „*microaveraged performance is somewhat misleading ... because more frequent topics are weighted heavier in the average* “ [92, Seite 327], und bevorzugen daher Macro-Averaging. Andere wiederum präferieren Micro-Averaging, da einzelne Themengebiete (Topics) entsprechend ihrer Auftittshäufigkeit bewertet werden. Bei der in dieser Arbeit vorliegenden Auswertung wurde Micro-Averaging verwendet.

Die Kennzahlen Precision und Recall werden häufig verwendet, um die Performanz von IR Algorithmen anzugeben. Allerdings werfen neue Studien ebenfalls auch Probleme dieser Kennzahlen auf [46, 87, 85, 93].

Zum einen bedarf eine genaue Schätzung des Recalls für eine Textgruppierung ein detailliertes Wissen über alle Dokumente einer Kollektion. Bei großen Kollektionen, wie sie heute verwendet werden, ist gerade dieses Wissen nicht vorhanden, wodurch eine genaue Angabe des Recalls nicht möglich ist. Zum anderen sind Recall und Precision zwei verwandte Kennzahlen, welche sich mit unterschiedlichen Aspekten des Ergebnisses auseinandersetzen. In vielen Situationen kann jedoch der Einsatz einer Kombination von Recall und Precision angebrachter sein.

Weiters bestimmen Recall und Precision die Effektivität eines Systems über einer Menge von Dokumentzuweisungen, die im Batch-Modus verarbeitet wurden. Bei neueren (IR) Systemen stellt allerdings die Interaktivität den Schlüssel zu einem guten Ergebnis dar. In solchen Fällen sind Kennzahlen, die den Informationsgehalt eines (IR) Prozesses quantifizieren, besser geeignet. Darüberhinaus sind Recall und Precision einfach zu definieren, wenn eine lineare Ordnung auf der Menge der bezogenen Dokumente gefordert wird. Für Systeme, die allerdings auf einer schwachen

Ordnung der Dokumente arbeiten, gelten Recall und Precision nicht als geeignete Kennzahlen, um deren Effektivität zu bestimmen [4]. Die Metriken Accuracy und Error werden im Bereich der Textgruppierung nur selten verwendet. Yang [93] zeigte, dass die typischerweise große Zahl im Nenner der Berechnungsformel diese Kennzahlen unempfindlich gegenüber der Anzahl richtiger Entscheidungen des Systems macht (im Gegensatz zu Recall und Precision).

### 6.2.2 Alternative Kennzahlen

Da die vorher genannten Kennzahlen nicht immer angebrachte Auswertungszahlen liefern, wurden in den letzten Jahren neue Alternativen entwickelt. Die Bekanntesten stellen dabei Kombinationen aus Recall und Precision dar.

Eine dieser Kombinationen stellt das F-Measure dar. Diese Metrik erlaubt es dem Benutzer, den Einfluss von Recall und Precision individuell zu bestimmen. Das F-Measure [6, 88, 81, 71, 38, 54, 16, 26, 96, 17, 93, 60, 1] berechnet sich über die Formel

$$\begin{aligned} F_\beta &= 1 - E_\beta \\ E_\beta &= 1 - \frac{(\beta^2+1)RP}{\beta^2 P + R} \end{aligned} \tag{6.7}$$

$P$ steht hierbei für die Precision, $R$ für den Recall. $\beta$ ist eine benutzerdefinierte Variable, die den Einfluß von Recall und Precision steuert und Werte aus dem Intervall $[0, \infty]$ annimmt. $F_0$ entspricht der Precision, $F_\infty$ dem Recall. Nimmt $\beta$ den Wert 1 an, wird Recall und Precision dieselbe Bedeutung zugemessen. Werte kleiner 1 steigern den Einfluß von Precision, während größere Werte dem Recall mehr Bedeutung zumessen.

Eine andere Kombination von Recall und Precision ist das Harmonic-Mean F [91], dass über die Formel

$$F(j) = \frac{2}{\frac{1}{R(j)} + \frac{1}{P(j)}} \tag{6.8}$$

berechnet wird. $R(j)$ stellt dabei den Recall der $j$-ten Kategorie, $P(j)$ die Precision des $j$-ten Kategorie dar. Das Ergebnis, $F(j)$, ist das harmonische Mittel von $R(j)$ und $P(j)$. Das Funktionsergebnis liegt im Intervall $[0, 1]$. Eine 0 bedeutet, dass keine relevanten Dokumente richtig zugewiesen wurden, eine 1, dass alle zugewiesenen Dokumente richtig sind. Weiters ergibt das Harmonic- Mean F nur hohe Werte, wenn sowohl Recall als auch Precision hoch sind. Durch die Bestimmung des Maximums der Funktion kann auf diese Weise der beste Kompromiss zwischen Recall und Precision gefunden werden.

Ein anderes Maß zur Bestimmung der Performanz (von IR Systemen) wurde von van Rijsbergen vorgeschlagen. Das sogenannte E (Evaluation) Measure [88] stellt ebenfalls eine Kombination aus Recall und Precision dar. Die Idee dahinter ist, das der Benutzer ebenfalls selbst angeben kann, ob mehr Wert auf Recall oder Precision gelegt werden soll. Das E Measure ist definiert als

$$E(j) = 1 - \frac{1 + b^2}{\frac{b^2}{R(j)} + \frac{1}{P(j)}} \tag{6.9}$$

wobei $R(j)$, wie schon beim Harmonic-Mean F, dem Recall und $P(j)$ der Precision entspricht. $b$ ist ein Benutzerparameter, der die relative Gewichtung von Recall und Precision bestimmt. Nimmt $b$ den Wert 1 an, arbeitet das E Measure exakt wie das Harmonic-Mean F. Größere Werte von $b$ bedeuten eine stärkere Einflußnahme der Precision, Werte für $b$ kleiner 1 bedeuten ein stärkeres Interesse am Recall.

Eine weitere Möglichkeit stellt die Berechnung des Breakeven Points (BEP) [52, 81, 82, 41, 40, 93, 55, 1] dar, dem Punkt an dem Recall und Precision denselben Wert annehmen. Meistens wird dazu sowohl der Recall als auch die Precision interpoliert, um einen Schnittpunkt berechnen zu können. Schapire et al. [77] zeigten jedoch auch Schwächen dieser Methoden auf.

Eine andere Methode ist die Berechnung der 11-Point Average Precision [81, 1]. Hierbei wird die Precision-Kurve für die Werte 0.0, 0.1, ..., 0.9 und 1.0 (über eine Interpolation) berechnet. Das Ergebnis wird durch die Bildung des Durchschnitts dieser 11 Werte berechnet.

Eine weitere wichtige Entscheidung fällt auf die Art der Bestimmung, wann eine

Clusterzuweisung nun korrekt ist. Nachdem die Clusterbildung anhand der Trainingsbeispiele stattgefunden hat, wird versucht, jedem Cluster eine oder mehrere (menschliche) Kategorie(n) zugewiesen. Auf diese Art kann festgestellt werden, ob ein Dokument in einem Cluster korrekt zugeordent wurde. Um die Kategorie eines Clusters zu bestimmen, gibt es mehrere Möglichkeiten:

- Es kann ein Grenzwert festgelegt werden, ab wann ein Cluster einer Kategorie entspricht. Dieser Grenzwert kann sowohl absolut (Anzahl von Dokumenten derselben Kategorie) oder relativ (prozentuelle Angabe von Dokumenten derselben Kategorie) erfolgen. Durch diese Festlegung kann ein Cluster auch mehreren, einer oder keiner Kategorie(n) entsprechen. Solche Grenzwerte werden festgelegt, um ein Mindestmaß an Übereinstimmung von Clustern mit vordefinierten Kategorien vorzugeben. Bei zu kleinen Grenzwerten oder großen, heterogenen Clustern kann dies zu sehr vielen Kategorieübereinstimmungen führen. Deshalb müssen diese Werte in Abhängigkeit des verwendeten Corpus und dessen Größe gewählt werden.

- Majority voting stellt eine zweite Möglichkeit dar. Hierbei entspricht ein Cluster lediglich dann einer Kategorie, wenn eine Kategorie der Dokumente am stärksten vertreten ist. Dadurch entspricht ein Cluster entweder genau einer oder keiner (Gewinner-)Kategorie. Kleine Cluster werden durch diese Art der Beurteilung meist durch keine Kategorie repräsentiert (im Falle zweier Dokumente unterschiedlicher Kategorien). Diese Art der Bewertung von Clustern scheint sehr plausibel zu sein, da ein Cluster entweder genau einer menschlichen Kategorie entspricht, oder es sich um einen Mischcluster handelt, der keiner Kategorie eindeutig zugewiesen werden kann.

In dieser Evaluation wurde das Majority voting verwendet, um die Gewinnerkategorie eines Clusters zu bestimmen. Dies entspricht einer pessimistischen Beurteilung des Ergebnisses, da Cluster mit wenigen Dokumenten oftmals keine Gewinnerkategorie aufweisen. Da die Auswertungen jedoch miteinander verglichen werden, fließt dieser Aspekt in beide Evaluationen gleichermaßen ein und kann somit vernachlässigt werden.

## 6.3 Das Corpus

Die Dokumentsammlung, das Corpus, stellt bei der Evaluation die Grundvoraussetzung dar. Da die Texte von Textgruppierungssystemen in Kategorien (Cluster) eingeteilt werden, ist es notwendig, dass die Dokumente des Corpus ebenfalls einer Kategorisierung unterliegen. Diese Vor-Kategorisierung wird im Test als Ideal-Referenzmodell angesehen, mit der das Ergebnis des Systems verglichen wird. Die Problematik einer bereits durch Menschen vorgenommenen Fehlkategorisierung bleibt natürlich bestehen, da sich oftmals auch Experten uneinig über bestimmte Kategorisierungen sind.

Im Internet sind einige Text-Corpora vorhanden, die sich zur Evaluation von IR Systemen eignen. Beispiele für im Netz frei erhältliche Text-Corpora sind das TIGer Corpus[1], das NeGra Corpus[2], das COSMAS-I[3] und COSMAS-II[4] Corpus, das Münster Tagging Project (MTP), das ECI-ELSNET Italian & German tagged subcorpus, das Karl-May-Corpus, das MULTEXT JOC Corpus (Multilingual Text Tools and Corpora)[5], die MLCC - Multilingual and Parallel Corpora[6] und das GeFRePaC - German French Reciprocal Parallel Corpus[7]. Leider sind diese Corpora nicht vorkategorisiert, weshalb sie für eine Evaluation von SyRS ungeeignet sind. Weitere Informationen zu deutschsprachigen Corpora des IDS sind ebenfalls im Web[8] zu finden.

Lediglich das GIRT-Corpus (German Indexing and Retrieval Testdatabase)[9] des Cross Language Evaluation Forum (CLEF)[10] kommt für diese Zwecke zum Einsatz. Leider war dieses im Zeitraum der Arbeit nicht verfügbar. Genauere Informationen über den Inhalt und Aufbau von GIRT kann aus [44] entnommen werden.

Aus diesem Grund wurde ein eigenes Corpus angelegt. Als Kandidaten wurden

[1] http://www.ims.uni-stuttgart.de/projekte/TIGER/ (Stand: 2003.09.08)
[2] http://www.coli.uni-sb.de/sfb378/negra-corpus/ (Stand: 2003.09.08)
[3] http://www.ids-mannheim.de/kt/projekte/cosmas_I/ (Stand: 2003.09.08)
[4] http://www.ids-mannheim.de/kt/corpors.shtm (Stand: 2002.03.03)
[5] http://www.lpl.univ-aix.fr/projects/multext/CORP/MUL4.de.html (Stand: 2003.09.08)
[6] http://www.hcrc.ed.ac.uk/Site/MLCC.html (Stand: 2003.09.08)
[7] http://www.solaris3.ids-mannheim.de/gefrepac.html (Stand: 2002:02.02)
[8] http://www.ids-mannheim.de/kt/corpora.html (Stand: 2003.09.08)
[9] http://clef.iei.pi.cnr.it:2002/girt-new1.html (Stand: 2002.02.02)
[10] http://clef.iei.pi.cnr.it:2002/ (Stand:2002.02.02)

gängige Datenarchive aus dem Web herangezogen, die ihre Dokumente in eine Kategorienstruktur eingebunden haben. In Betracht gezogen wurde unter anderem die FutureZone[11], die Kryptocrew[12], die Süddeutsche Zeitung[13] oder der Standard[14].

Auf der Basis von Diplomica.com[15] wurde letztendlich selbstständig ein Corpus entworfen. Diplomica.com bot sich an, da die Texte verhältnismäßig einfach zu beziehen und bereits eine geeignete Kategorienstruktur vorlag. Zum damaligen Stand (2002.10.15) befanden sich ca. 6000 Dokumente, davon 95% deutsche Texte, aus über 400 Kategorien auf dieser WebSite. Die Kategorien waren hierarchisch in bis zu 5 Subklassifikationsebenen organisiert. Es existierten 28 Hauptkategorien wie „Biologie“, „Mathematik“, „Architektur / Raumplanung“ oder die „Betriebswirtschaft - Funktional“. Diese Hauptgruppen waren wiederum unterteilt in verschiedene Unterkategorien. Die funktionale Betriebswirtschaftslehre umfaßte beispielsweise 21 Themenblöcke wie Controlling, Marketing oder Organisation. Der Punkt Organisation wiederum teilte sich auf in 12 Untergruppen wie Workflow oder Produktion. Jede Diplomarbeit war genau einer dieser Kategorien zugewiesen. Einige Kategorien waren aufgrund ihrer speziellen Unterklassifikation mit nur wenigen Dokumenten belegt. Trotzdem lag der Durchschnitt bei ca. 12 Dokumenten pro Kategorie. Die Diplomarbeiten wurden mit Titel, Untertitel, AutorIn, Umfang, Hochschule, Art der Arbeit, Abgabetermin, Note, Preis, Bestellnummer, Sprache, Medien, Inhaltsangabe und Inhaltsverzeichnis spezifiziert.

Die aus dem Netz bezogenen Dateien lagen im HTML-Format vor. In einem ersten Schritt wurden die HTML-Tags entfernt, um die Texte für SyRS aufzubereiten. Anschließend wurde lediglich die Inhaltsangabe extrahiert und in einer eigenständigen Textdatei gespeichert, um SyRS als Input zu dienen. Durchschnittlich lag die Länge eines Textes bei etwa 250 Wörtern. Auf andere Datenfelder wie den Titel wurde in dieser Arbeit verzichtet. Die Kategorie wurde in Form einer fortlaufenden Nummer im Dateinamen festgehalten.

Aus den zur Verfügung stehenden Dokumenten wurden zwei Textcorpora gebildet.

[11] http://futurezone.orf.at/ (Stand: 2003.09.08)
[12] http://www.kryptocrew.de/home/ (Stand: 2003.09.08)
[13] http://www.sueddeutsche.de/ (Stand: 2003.09.08)
[14] http://derstandard.at/ (Stand: 2003.09.08)
[15] http://www.diplomica.com (Stand: 2002.10.15)

Die Dokumente wurden dabei aus zufällig ermittelten Blatt-Kategorien bezogen, wobei keine Rücksicht auf deren Schachtelungstiefe genommen und alle Kategorien einander gleichgestellt wurden. Corpus 1 setzte sich dabei aus einem Querschnitt des gesamten Diplomica-Programms zusammen. Corpus 2 wurde speziell aus dem Bereich der funktionalen Betriebswirtschaftslehre gewählt. Die im Verhältnis zum Gesamtumfang geringen Corpusgrößen resultieren aus den Obergrenzen der Termvektorlängen und der dadurch entstehenden Thema-Rhema Matrix (8600 x 3500). Beide Corpora wurden ebenfalls nach dem Zufallsprinzip in eine Trainings- (60 %) und eine Testkollektion (40 %) geteilt (siehe Tabelle 6.2). Basierend auf diesen Daten wurde die Evaluation durchgeführt.

Tabelle 6.2: Übersicht über die gebildeten Corpora

| **Bezeichnung** | | **# Dokumente** | **# Kategorien** | **avg. D/K** | **# Wörter** |
|---|---|---|---|---|---|
| Corpus 1 | Training | 252 | 80 | 3.15 | 59108 |
| | Testing | 167 | 66 | 2.53 | 39005 |
| | Gesamt | 418 | 96 | 4.35 | 98113 |
| Corpus 2 | Training | 207 | 38 | 5.45 | 47094 |
| | Testing | 132 | 32 | 4.13 | 29334 |
| | Gesamt | 339 | 41 | 8.27 | 76428 |

## 6.4 Ergebnisse der Experimente

Mit den in Tabelle 6.2 vorgestellten Corpora und den Kennzahlen aus Abschnitt 6.2 wurde SyRS ausgewertet. In diesem Abschnitt wird auf die verwendeten Parametereinstellungen der NN und die erzielten Ergebnisse während des Trainings und des Testens eingegangen.

### 6.4.1 Experimentelle Methodik

Alle vorgestellten Ergebnisse wurden auf den in Tabelle 6.2 angeführten Corpora durchgeführt. Begonnen wurde mit der Erstellung der Dokumentrepräsentationen

der einzelnen Corpora (Trainings- und Testdokumente) durch das Natural Language Modul. Anschließend wurde das Dokumentclustering und das Clusterretrieval vom Neuronalen Netzwerk Modul durchgeführt.

Wie schon zuvor angesprochen, besteht der Arbeitszyklus Neuronaler Netze aus zwei Phasen. Während der Lernphase wurden die beiden selbstlernenden Einheiten, das FAM Netz und das fuzzy ART Netz, anhand der Testdokumente trainiert. Die Dokumente wurden dazu in namentlicher Reihenfolge an das Neuronale Netzwerk Modul übergeben. Zuerst wurde das vorgeschaltete FAM Netz (des Thema-Rhema Modells) trainiert. Im Anschluß daran fand die eigentliche Clusterbildung anhand des ART Netzes statt. Während des Trainings wurde jedes Dokument genau einem Cluster zugewiesen, wobei es in die Bildung des jeweiligen Clusterzentrums einfloß.

Die Zeitdauer des Trainings ist dabei von der Art der Netze, den gewählten Parametereinstellungen, der Anzahl von Dokumenten und deren Repräsentationform abhängig. Während das fuzzy ART Netz relativ schnell trainiert werden kann, bedarf das FAM Netz einiges mehr an Rechenleistung. Beispielsweise dauerte das Trainieren des ART Netzes mittels der heavy-Variante des Vektorenmodells ca. 4 Minuten pro Parametereinstellung für das Corpus 1. Die Kombination des FAM und ART Netzes hingegen, wie sie vom Thema-Rhema Modell verwendet wird, rechnete ca. 20 Stunden auf denselben Daten. Dieses Ergebnis wurde auf einem Pentium4 mit 1700 MHz und 256 MB Hauptspeicher ermittelt, der mit Debian-Linux betrieben wurde.

Die zweite Phase, das Testen bzw. die Retrievalphase, schloß direkt an das Training an. Wiederum wurden die Testdokumente nach Namen sortiert dem System übergeben. Jedes Dokument wurde mit allen in SyRS vorhandenen Clusterzentren verglichen. War ein Cluster entsprechend ähnlich (in Bezug auf den Parameter $\rho$), so wurde dies vermerkt. War ein Dokument keinem Cluster ähnlich, fand keine Zuweisung statt. Nachdem alle Testdokumente abgearbeitet waren, wurden diejenigen Dokumente an den Benutzer zurückgegeben, die in zum Dokument ähnlichen Clustern lagen. Das Testen beanspruchte im Verhältnis zum Training nur geringfügige Kapazitäten. Auf derselben Maschine dauerte das Testen von Corpus 1 mittels der heavy-Variante des Vektorenmodells ca. 2 Minuten. Die heavy Variante des Thema-Rhema Modells rechnete dazu ca. 8 Minuten.

Um den Erfolg des Thema-Rhema Modells mit dem des Vektorenmodells zu vergleichen, wurde das Retrieval-Ergebnis nach der Testphase anhand von Kennzahlen ermittelt. Als Vergleichsmaß der einzelnen Varianten beider Modelle wurde Recall, Precision und der Breakeven Point verwendet. Beim Vergleich des Verkorenmodells mit dem Thema-Rhema Modell wurde zusätzlich noch das F-Measure als Bewertungskriterium hinzugezogen. Den Verlauf der Kennzahlenentwicklung veranschaulichen die beigefügten Grafiken. Verglichen wurden sowohl die einzelnen Werte der Kennzahlen als auch deren Entwicklung anhand der getätigten Parametereinstellungen. Besonders von Interesse ist der Bereich der Parametereinstellungen, in denen die durchschnittliche Clustergröße den Bereichen der tatsächlichen Kategoriengröße des Trainingscorpus nahekommt.

### 6.4.2 Parametereinstellungen

Die Parametereinstellungen der NN spielen eine wichtige Rolle für deren Arbeitsergebnis und Performanz. Da in SyRS zwei verschiedenartige NN zum Einsatz kommen, werden hier die Grundeinstellungen beider Netztypen vorgestellt. Wie bereits im Kapitel 5 vorgestellt, wird das erste Neuronale Netz, das Fuzzy Associative Memory (FAM), nur von einem Parameter gesteuert, der Schrittgröße $\eta$. Als Grundeinstellung wurde hier für alle Testläufe 0.08 verwendet. $\eta$ kann Werte aus dem Intervall [0,1] annehmen. Größere Werte für $\eta$ führen zu einer schnelleren Konvergenz des NNs, jedoch auch zur Bildung lokaler Maxima. Kleinere Werte hingegen führen zu einer größeren Anzahl von Berechnungen und somit zu längeren Arbeitszyklen des NNs.

Das zweite NN, das fuzzy Adaptive Resonance Theory Netz (fuzzy ART), wird von drei Parametern beeinflußt. $\alpha$, der sogenannte Choice Parameter, ist auf 0.05 eingestellt. Dies wird im Verlauf der Experimente nicht geändert. $\beta$ ist der Lernparameter, der während dieser Experimente auf 1 gesetzt wurde. Dies erlaubt es dem Netz sehr schnell zu lernen. Zusätzlich wird das fuzzy ART Netz noch durch den Parameter $\rho$ gesteuert. Er gibt an, inwieweit ein Clusterprototyp mit einem Dokument übereinstimmen muss, um das Dokument diesem Cluster zuweisen zu können. Dieser Parameter wurde in mehrfachen Testgängen für die Werte 0.00, 0.01, ..., 0.09, 0.10,

0.15, 0.20, ..., 0.95 und 0.99 untersucht. Je größer der Parameter $\rho$ gewählt wird, desto ähnlicher müssen Dokumente innerhalb eines Clusters sein. Ein großer Wert von $\rho$ bedeutet eine größere Anzahl kompakterer Cluster. Im Gegensatz dazu führt ein kleinerer Wert von $\rho$ zur Bildung einer geringeren Anzahl größerer Cluster.

### 6.4.3 Das Training

Während des Trainings findet die Clusterbildung statt, wobei ähnliche Dokumente in denselben Clustern zusammengefaßt werden. Dies geschieht anhand der identifizierten Indexterme, die während der Textanalyse ermittelt wurden. Die Größenordnung der Indexterme für die einzelnen Corpora veranschaulicht Tabelle 6.3.

Tabelle 6.3: Analyseergebnisse der Corpora

| **Bezeichnung** | | **Vektorenmodell** | | **Thema-Rhema Analyse** | | | |
|---|---|---|---|---|---|---|---|
| | | **light** | **heavy** | **light** | | **heavy** | |
| | | **# Vek** | **# Vek** | **# Th** | **#Rh** | **# Th** | **#Rh** |
| Corpus 1 | Training | 7642 | 9260 | 1633 | 6961 | 3486 | 8598 |
| | Testing | 5498 | 6729 | 1146 | 4974 | 2550 | 6235 |
| Corpus 2 | Training | 5328 | 6446 | 1215 | 4769 | 2542 | 5933 |
| | Testing | 3918 | 4783 | 803 | 3542 | 1845 | 4415 |

In Tabelle 6.3 ist zu beachten, dass die Summe der Themata und Rhemata größer ist als die Anzahl der entsprechenden Indexterme des Vektorenmodells. Da es sich um dieselben Corpora handelt, bei denen die Indexterme auf dieselbe Weise bestimmt wurden, ist dies dadurch zu erklären, dass gleichgeschriebene Wörter sowohl als Thema als auch (gleichzeitig im selben Text) als Rhema vorkommen können. In der Liste der Themata und Rhemata werden gleiche Wörter unter demselben Begriff zusammengefaßt, die Einträge der Listen selbst können jedoch nicht miteinander in Verbindung gebracht werden. Zusätzlich zeigt die Tabelle einen eingeschränkteren Wortschatz des Corpus 2 auf. Da dieses Corpus einen speziellen Ausschnitt von Diplomica.com, die funktionale Betriebswirtschaftslehre, umfaßt (im Gegensatz zu

Corpus 1, der einen Querschnitt durch alle Themenbereiche beinhaltet), fällt dieses Ergebnis wie erwartet aus. Die durchschnittliche Kategoriengröße der Trainingsdaten liegt bei Corpus 1 bei 3.15, bei Corpus 2 bei 5.45 Dokumenten pro Kategorie.

Ein wichtiger Punkt während des Trainings stellt die Input-Reihenfolge der einzelnen Textdokumente dar. Da die Bildung des ersten Clusters mit dem ersten Dokument beginnt und alle weiteren Cluster aufgrund von unähnlichen Dokumenten zu den zuvor gebildeten Clustern generiert werden, ist diese Reihung der Trainingsdokumente bei kleinem Datenumfang (wie hier) entscheidend. Ein einmaliges Training mittels einer festgelegten Menge von Dokumenten führt somit zur Bildung unterschiedlicher Clusterstrukturen. Dieses Problem kann durch ein andauerndes Weitertrainieren des Neuronalen Netzes, das solange durchgeführt wird, bis sich stabile Cluster bilden (die Clusterstruktur nicht mehr verändert), verhindert werden. Da in dieser Evaluation die Dateinamen der Corpusdokumente mit dem jeweiligen Kategoriekürzel beginnen und diese Dokumente in namentlicher Reihenfolge an das System übergeben werden, sollten sich die Cluster in einer günstigen Weise entwickeln können, da alle Dokumente derselben Kategorie hintereinander verarbeitet werden.

Die Tabellen 6.4 und 6.5 und die Abbildungen 6.1 und 6.2 fassen die Trainingsergebnisse des Vektoren- und des Thema-Rhema Modells zusammen. Ersichtlich ist die durchschnittliche Clustergröße (Dokumente / Kategorie) und die Anzahl der Cluster (# K) für jeden der zwei Corpora.

Das Ergebnis des auf dem Vektorenmodell (siehe Tabelle 6.4) basierenden Clusterings zeigt sowohl in Corpus 1 als auch in Corpus 2 deutlich, dass sich die einzelnen Dokumente generell nicht sehr ähnlich sind. Dies drückt sich in der großen Anzahl sehr kleiner Cluster aus, die bis zu sehr kleinen Ähnlichkeitsübereinstimmungen von $\rho$ entstehen. Werte von $\rho \geq 0.20$ führen zu Clustern, die jeweils nur ein einziges Dokument enthalten. Die Prototypen solcher Cluster entsprechen dem jeweiligen Dokument eins zu eins. Die durchschnittliche Clustergröße erreicht erst ab Werten von $\rho$ aus dem Bereich von 0.02 und 0.01 den der Trainingsdokumente. Die Ergebnisse der light- und der heavy- Variante variieren in beiden Corpora nur minimal. Daraus ist zu schließen, das sich der Einfluß von Adjektiven und Adverbien nicht im besonderen Maße auf die Clusterbildung auswirkt. Die gebildeten Cluster bei Corpus

2 sind geringfügig größer als die bei Corpus 1. Da Corpus 2 spezieller gewählt wurde und über ein eingeschränkteres Vokabular sowie einer höheren Durchschnittsanzahl an Dokumeten pro Kategorie verfügt, fällt dieses Ergebnis wie erwartet aus.

Tabelle 6.4: Trainingsergebnisse VSM (Corpus 1 & Corpus 2)

| $\rho$ | **Corpus 1** | | | | **Corpus 2** | | | |
|---|---|---|---|---|---|---|---|---|
| | **light** | | **heavy** | | **light** | | **heavy** | |
| | **# K** | **D/K** | **# K** | **D/K** | **# K** | **D/K** | **# K** | **D/K** |
| $\geqq 0.20$ | 252 | 1.00 | 252 | 1.00 | 207 | 1.00 | 207 | 1.00 |
| 0.15 | 252 | 1.00 | 251 | 1.00 | 199 | 1.04 | 202 | 1.02 |
| 0.10 | 235 | 1.07 | 240 | 1.05 | 172 | 1.20 | 175 | 1.18 |
| 0.09 | 229 | 1.10 | 232 | 1.09 | 159 | 1.30 | 162 | 1.27 |
| 0.08 | 213 | 1.18 | 218 | 1.16 | 151 | 1.37 | 149 | 1.39 |
| 0.07 | 194 | 1.30 | 203 | 1.24 | 135 | 1.53 | 133 | 1.56 |
| 0.06 | 171 | 1.47 | 176 | 1.43 | 121 | 1.71 | 121 | 1.71 |
| 0.05 | 153 | 1.65 | 153 | 1.65 | 108 | 1.92 | 108 | 1.92 |
| 0.04 | 136 | 1.85 | 134 | 1.88 | 106 | 1.95 | 103 | 2.01 |
| 0.03 | 123 | 2.05 | 125 | 2.02 | 97 | 2.13 | 97 | 2.13 |
| 0.02 | 112 | 2.25 | 113 | 2.23 | 85 | 2.41 | 88 | 2.35 |
| 0.01 | 76 | 3.32 | 82 | 3.07 | 57 | 3.63 | 58 | 3.57 |
| 0.00 | 1 | 252 | 1 | 252 | 1 | 207 | 1 | 207 |

Die Trainingsergebnisse des auf dem Thema-Rhema Modells (siehe Tabelle 6.5) basierenden Clusterings zeigen in beiden Corpora eine etwas größere Ähnlichkeit unter Dokumenten auf, da erst ab Werten von $\rho \geq 0.50$ einelementige Cluster gebildet werden. Dies ist bedingt durch ein Auslösen von neuen, nicht im Text genannten Termen (thematisches Material) durch explizit im Text genanntes (rhematisches) Material. Dadurch geschieht das Clustering anhand einer geringeren Anzahl von Indextermen.

Die durchschnittliche Clustergröße liegt in Corpus 1 im Bereich von $\rho = 0.09$, in Corpus 2 bei Werten von $\rho = 0.06$. Hier variiert die light- von der heavy-Variante

deutlich, wobei die light- Variante etwas größere Cluster bildet. Der zusätzliche Ausreißer des light-Thema-Rhema Modells bei $\rho = 0.15$ in beiden Corpora sollte diesbezüglich nicht überbewertet werden. Anscheinend ist hier eine Grenze erreicht, bei der mehrere Dokumente einander ähnlich genug sind (hinsichtlich $\rho$), um sie zusammenzufassen.

Der Einfluß von Adjektiven und Adverbien ist in diesem Modell deutlicher zu spüren, da diese Wortfamilien ebenfalls neues, thematisches Material anstoßen. Auch dieses Modell liefert bei Corpus 2 etwas größere Cluster als bei Corpus 1, wie es den Erwartungen entspricht.

Tabelle 6.5: Trainingsergebnisse Th-Rh (Corpus 1 & Corpus 2)

| $\rho$ | **Corpus 1** | | | | **Corpus 2** | | | |
|---|---|---|---|---|---|---|---|---|
| | **light** | | **heavy** | | **light** | | **heavy** | |
| | **# K** | **D/K** | **# K** | **D/K** | **# K** | **D/K** | **# K** | **D/K** |
| $\geqq 0.50$ | 252 | 1.00 | 252 | 1.00 | 207 | 1.00 | 207 | 1.00 |
| 0.45 | 251 | 1.00 | 252 | 1.00 | 206 | 1.00 | 207 | 1.00 |
| 0.40 | 249 | 1.01 | 252 | 1.00 | 203 | 1.02 | 207 | 1.00 |
| 0.35 | 245 | 1.03 | 251 | 1.00 | 198 | 1.05 | 207 | 1.00 |
| 0.30 | 235 | 1.07 | 249 | 1.01 | 189 | 1.10 | 202 | 1.04 |
| 0.25 | 218 | 1.16 | 239 | 1.05 | 173 | 1.20 | 192 | 1.08 |
| 0.20 | 188 | 1.34 | 219 | 1.15 | 147 | 1.41 | 167 | 1.24 |
| 0.15 | 89 | 2.83 | 173 | 1.46 | 66 | 3.14 | 138 | 1.50 |
| 0.10 | 89 | 2.83 | 110 | 2.29 | 66 | 3.14 | 89 | 2.33 |
| 0.09 | 78 | 3.23 | 96 | 2.63 | 64 | 3.23 | 75 | 2.76 |
| 0.08 | 68 | 3.71 | 83 | 3.04 | 56 | 3.70 | 65 | 3.18 |
| 0.07 | 58 | 4.34 | 69 | 3.65 | 48 | 4.31 | 61 | 3.39 |
| 0.06 | 50 | 5.04 | 59 | 4.27 | 40 | 5.18 | 52 | 3.98 |
| 0.05 | 46 | 5.48 | 53 | 4.75 | 37 | 5.59 | 43 | 4.81 |
| 0.04 | 43 | 5.86 | 43 | 5.86 | 35 | 5.91 | 35 | 5.91 |
| 0.03 | 40 | 6.30 | 39 | 6.46 | 33 | 6.27 | 35 | 5.91 |
| 0.02 | 40 | 6.30 | 37 | 8.81 | 32 | 6.47 | 30 | 6.90 |
| 0.01 | 37 | 6.81 | 28 | 9.00 | 32 | 6.47 | 30 | 6.90 |
| 0.00 | 1 | 252 | 1 | 252 | 1 | 207 | 1 | 207 |

Im direkten Vergleich des Vektorenmodells mit dem Thema-Rhema Modell führt derselbe Wert von $\rho$ beim Thema-Rhema Modell zur Bildung einer geringeren Anzahl größerer Cluster (siehe Abbildungen 6.1 und 6.2). Dies ist bedingt durch das Rhema-Thema Mapping, also das Abbilden von Rhemata auf Themata. Dadurch kommt es zu einer Reduktion des für das Clustering verwendeten Termmaterials (nur Themata). Da die Rhemata zusätzlich weitere Themata anstoßen, wird dieser

Effekt noch verstärkt. Der Anstieg der Kurven beider Modelle ist nahezu (bis auf die Ausreißer bei $\rho = 0.15$ des light-Thema-Rhema Modells) gleich, wird die Kurve des Vektorenmodells nach rechts verschoben.

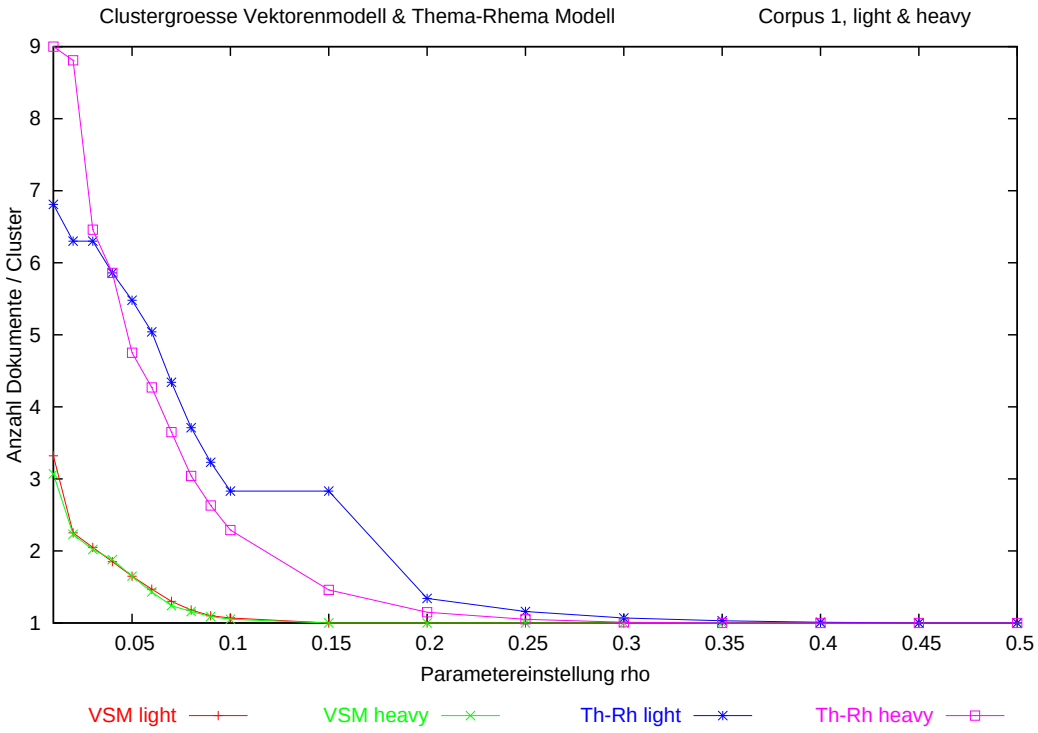



Abbildung 6.1: Clustergröße - Vergleich Corpus 1

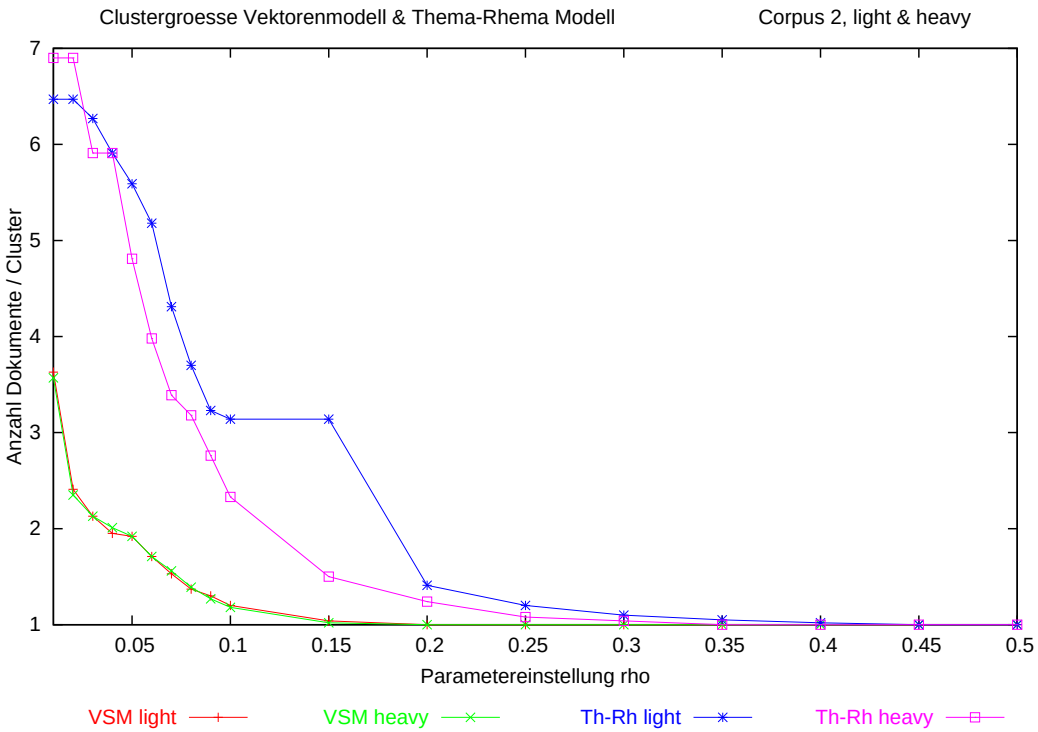



Abbildung 6.2: Clustergröße - Vergleich Corpus 2

### 6.4.4 Das Testen

Während des Testens wurden die Kennzahlen Recall (R), Precision (P), Fallout (F), Accuracy (A) und Error (E) ermittelt. Die folgenden Tabellen (6.6 bis 6.9) und Abbildungen (6.3 bis 6.14) zeigen den Verlauf dieser Kennzahlen hinsichtlich des Parameters $\rho$. Die Diskussion beschränkt sich jedoch auf die Bewertung von Recall, Precision und dem Breakeven Point.

Tabelle 6.6: Testergebnis VSM, Corpus 1

| $\rho$ | **Corpus 1 - light** | | | | | **Corpus 1 - heavy** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **R** | **P** | **F** | **A** | **E** | **R** | **P** | **F** | **A** | **E** |
| 0.50 | 0.014 | 1.000 | 0.000 | 0.138 | 0.862 | 0.014 | 1.000 | 0.000 | 0.138 | 0.862 |
| 0.45 | 0.014 | 1.000 | 0.000 | 0.138 | 0.862 | 0.014 | 1.000 | 0.000 | 0.138 | 0.862 |
| 0.40 | 0.014 | 1.000 | 0.000 | 0.138 | 0.862 | 0.014 | 1.000 | 0.000 | 0.138 | 0.862 |
| 0.35 | 0.014 | 1.000 | 0.000 | 0.138 | 0.862 | 0.014 | 1.000 | 0.000 | 0.138 | 0.862 |
| 0.30 | 0.014 | 1.000 | 0.000 | 0.138 | 0.862 | 0.014 | 1.000 | 0.000 | 0.138 | 0.862 |
| 0.25 | 0.014 | 0.500 | 0.087 | 0.138 | 0.862 | 0.014 | 0.667 | 0.045 | 0.138 | 0.862 |
| 0.20 | 0.014 | 0.500 | 0.087 | 0.138 | 0.862 | 0.014 | 0.500 | 0.087 | 0.138 | 0.862 |
| 0.15 | 0.014 | 0.333 | 0.160 | 0.137 | 0.863 | 0.022 | 0.273 | 0.276 | 0.143 | 0.857 |
| 0.10 | 0.629 | 0.109 | 0.986 | 0.112 | 0.888 | 0.586 | 0.092 | 0.985 | 0.098 | 0.902 |
| 0.09 | 0.928 | 0.100 | 0.997 | 0.102 | 0.898 | 0.866 | 0.102 | 0.997 | 0.103 | 0.897 |
| 0.08 | 1.000 | 0.084 | 0.999 | 0.085 | 0.915 | 0.992 | 0.079 | 1.000 | 0.079 | 0.921 |
| 0.07 | 1.000 | 0.067 | 1.000 | 0.067 | 0.933 | 1.000 | 0.072 | 1.000 | 0.072 | 0.928 |
| 0.06 | 1.000 | 0.057 | 1.000 | 0.057 | 0.943 | 1.000 | 0.051 | 1.000 | 0.051 | 0.949 |
| 0.05 | 1.000 | 0.046 | 1.000 | 0.046 | 0.954 | 1.000 | 0.044 | 1.000 | 0.044 | 0.956 |
| 0.04 | 1.000 | 0.043 | 1.000 | 0.043 | 0.957 | 1.000 | 0.039 | 1.000 | 0.039 | 0.961 |
| 0.03 | 1.000 | 0.039 | 1.000 | 0.039 | 0.961 | 1.000 | 0.034 | 1.000 | 0.034 | 0.966 |
| 0.02 | 1.000 | 0.035 | 1.000 | 0.035 | 0.965 | 1.000 | 0.034 | 1.000 | 0.034 | 0.966 |
| 0.01 | 1.000 | 0.042 | 1.000 | 0.042 | 0.958 | 1.000 | 0.036 | 1.000 | 0.036 | 0.964 |
| 0.00 | 1.000 | 0.047 | 1.000 | 0.047 | 0.953 | 1.000 | 0.008 | 1.000 | 0.008 | 0.992 |

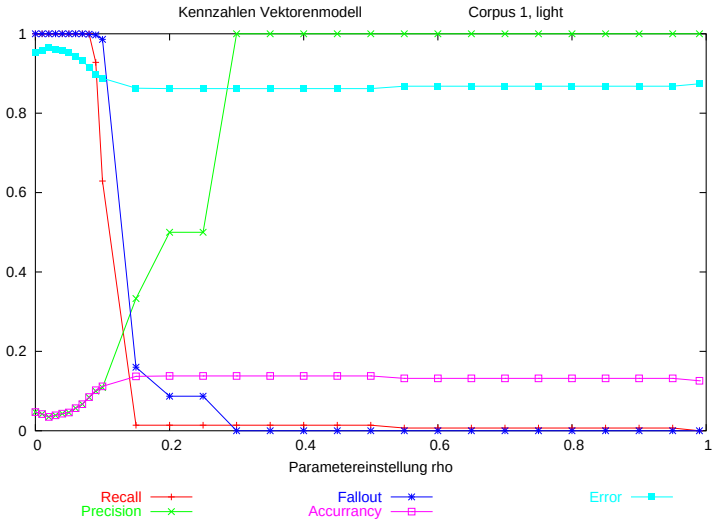



Abbildung 6.3: VSM light, Corpus 1

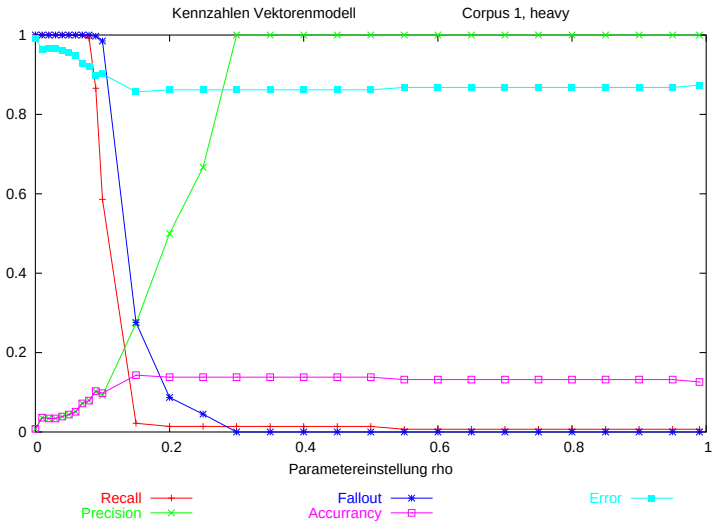



Abbildung 6.4: VSM heavy, Corpus 1

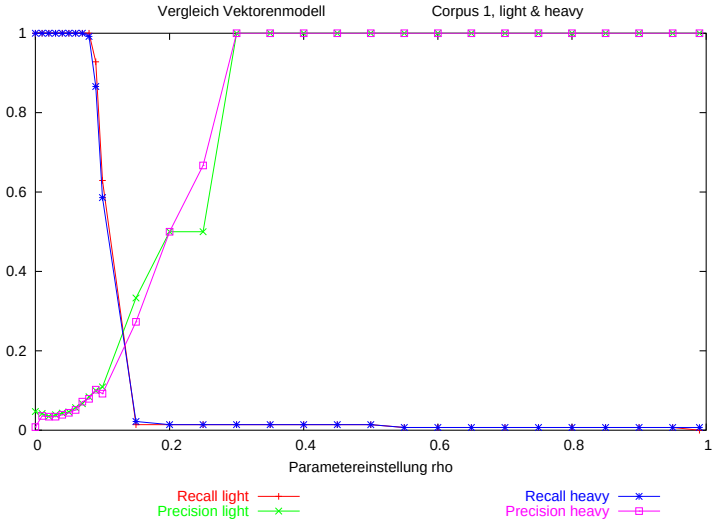



Abbildung 6.5: VSM light vs. heavy, Corpus 1

In Corpus 1 ist der Unterschied zwischen der light- und der heavy-Variante des Vektorenmodells (siehe Tabelle 6.6 und Abbildungen 6.3, 6.4 und 6.5) kaum bemerkbar. Die light-Variante übertrifft die heavy-Variante in ihren Recall- und Precision-Werten minimal (bis auf die Precision-Ausnahme bei 0.25). Auch hier ist das zusätzliche Wortmaterial (Adjektive und Adverbien) der heavy-Variante nur unmerklich spürbar. Der Breakeven Point der light-Variante liegt mit 0.23 ($\rho = 0.14$) leicht über dem der heavy-Variante mit 0.21 ($\rho = 0.14$), wie Tabelle 6.10 zeigt.

Die interessanten Werte von $\rho$ (gleiche durchschnittliche Clustergröße) liegen bei beiden Varianten um 0.01, das zwar einen Recall von 1 bedeutet, jedoch die Precision auf unter 0.05 drückt. Werte von $\rho \leq 0.30$, bei dem nur einelementige Cluster existieren, ist der Recall auf 0.014 gefallen, die Precision jedoch auf 1 angestiegen. Daraus ist die Gegenentwicklung von Recall und Precision schön zu erkennen.

Tabelle 6.7: Testergebnis SVM, Corpus 2

| $\rho$ | **Corpus 2 - light** | | | | | **Corpus 2 - heavy** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **R** | **P** | **F** | **A** | **E** | **R** | **P** | **F** | **A** | **E** |
| 0.50 | 0.047 | 0.750 | 0.500 | 0.061 | 0.939 | 0.047 | 0.750 | 0.500 | 0.061 | 0.939 |
| 0.45 | 0.047 | 0.750 | 0.500 | 0.061 | 0.939 | 0.047 | 0.750 | 0.500 | 0.061 | 0.939 |
| 0.40 | 0.047 | 0.750 | 0.500 | 0.061 | 0.939 | 0.047 | 0.750 | 0.500 | 0.061 | 0.939 |
| 0.35 | 0.047 | 0.750 | 0.500 | 0.061 | 0.939 | 0.047 | 0.750 | 0.500 | 0.061 | 0.939 |
| 0.30 | 0.047 | 0.750 | 0.500 | 0.061 | 0.939 | 0.047 | 0.750 | 0.500 | 0.061 | 0.939 |
| 0.25 | 0.047 | 0.750 | 0.500 | 0.061 | 0.939 | 0.047 | 0.750 | 0.500 | 0.061 | 0.939 |
| 0.20 | 0.055 | 0.778 | 0.500 | 0.068 | 0.932 | 0.055 | 0.778 | 0.500 | 0.068 | 0.932 |
| 0.15 | 0.205 | 0.571 | 0.947 | 0.184 | 0.816 | 0.152 | 0.436 | 0.957 | 0.133 | 0.867 |
| 0.10 | 0.984 | 0.238 | 0.998 | 0.238 | 0.762 | 0.988 | 0.221 | 0.998 | 0.222 | 0.778 |
| 0.09 | 1.000 | 0.230 | 1.000 | 0.230 | 0.770 | 0.996 | 0.199 | 0.999 | 0.199 | 0.801 |
| 0.08 | 1.000 | 0.198 | 1.000 | 0.198 | 0.802 | 1.000 | 0.171 | 1.000 | 0.171 | 0.829 |
| 0.07 | 1.000 | 0.176 | 1.000 | 0.176 | 0.824 | 1.000 | 0.173 | 1.000 | 0.173 | 0.827 |
| 0.06 | 1.000 | 0.164 | 1.000 | 0.164 | 0.836 | 1.000 | 0.157 | 1.000 | 0.157 | 0.843 |
| 0.05 | 1.000 | 0.142 | 1.000 | 0.142 | 0.858 | 1.000 | 0.148 | 1.000 | 0.148 | 0.852 |
| 0.04 | 1.000 | 0.134 | 1.000 | 0.134 | 0.866 | 1.000 | 0.122 | 1.000 | 0.122 | 0.878 |
| 0.03 | 1.000 | 0.125 | 1.000 | 0.125 | 0.875 | 1.000 | 0.121 | 1.000 | 0.121 | 0.879 |
| 0.02 | 1.000 | 0.118 | 1.000 | 0.118 | 0.882 | 1.000 | 0.121 | 1.000 | 0.121 | 0.879 |
| 0.01 | 1.000 | 0.149 | 1.000 | 0.149 | 0.851 | 1.000 | 0.159 | 1.000 | 0.159 | 0.841 |
| 0.00 | 1.000 | 0.164 | 1.000 | 0.164 | 0.836 | 1.000 | 0.212 | 1.000 | 0.212 | 0.788 |

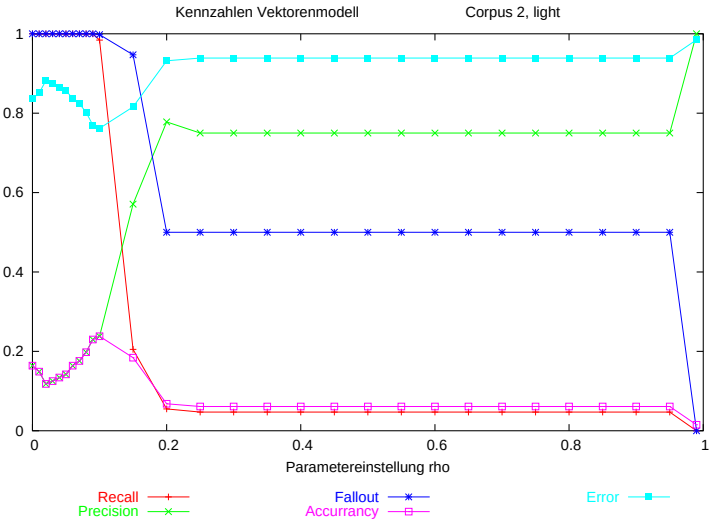



Abbildung 6.6: VSM light, Corpus 2

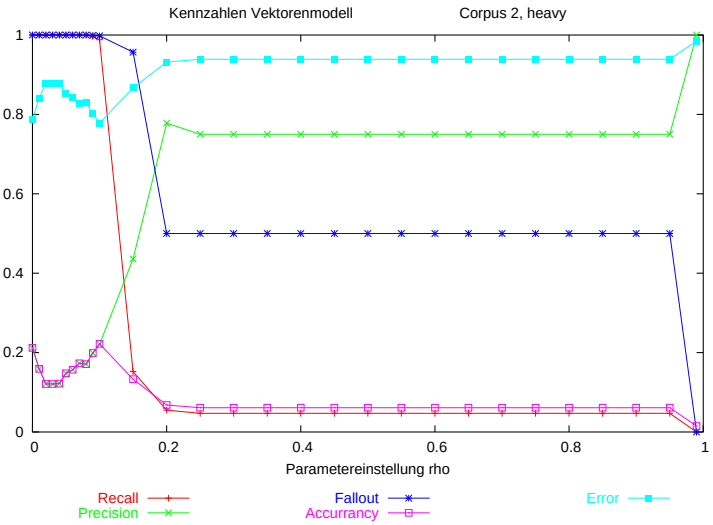



Abbildung 6.7: VSM heavy, Corpus 2

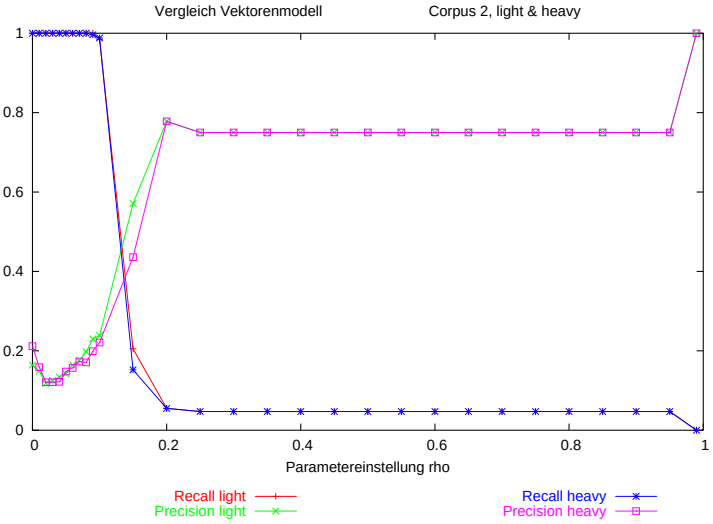



Abbildung 6.8: VSM light vs. heavy, Corpus 2

In Corpus 2 sind die Auswirkungen von Adjektiven und Adverbien bei der heavy-Variante des Vektorenmodells (siehe Tabelle 6.7 und Abbildungen 6.6, 6.7 und 6.8) ebenfalls nur gering spürbar. Wiederum liegen die Werte der light-Variante leicht über denen der heavy-Variante. Die zusätzlichen Indexterme der heavy-Variante scheinen auch hier keinen besonderen Einfluß auf das Ergebnis zu haben. Dies wird ebenfalls durch den Breakeven Point bestätigt, der bei der light-Variante mit 0.47 ($\rho = 0.13$) und bei der heavy-Variante mit 0.38 ($\rho = 0.14$) angeschrieben ist (siehe Tabelle 6.10).

Die durchschnittliche Clustergröße liegt auch hier bei Werten von $\rho \leq 0.01$. Der Recall-Wert nimmt an dieser Stelle wiederum den Wert 1 an, die Precision liegt jedoch unter 0.22. Im Bereich von $\rho \geq 0.25$ (nur einelementige Cluster) liegt der Recall bei 0.047, die Precision bei 0.75.

Tabelle 6.8: Testergebnis Th-Rh, Corpus 1

| $\rho$ | **Corpus 1 - light** | | | | | **Corpus 1 - heavy** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **R** | **P** | **F** | **A** | **E** | **R** | **P** | **F** | **A** | **E** |
| 0.50 | 0.007 | 0.071 | 0.433 | 0.108 | 0.892 | 0.007 | 0.100 | 0.321 | 0.120 | 0.880 |
| 0.45 | 0.008 | 0.045 | 0.583 | 0.096 | 0.904 | 0.008 | 0.059 | 0.471 | 0.114 | 0.886 |
| 0.40 | 0.025 | 0.083 | 0.702 | 0.101 | 0.899 | 0.008 | 0.043 | 0.550 | 0.114 | 0.886 |
| 0.35 | 0.074 | 0.114 | 0.838 | 0.110 | 0.890 | 0.009 | 0.029 | 0.667 | 0.107 | 0.893 |
| 0.30 | 0.193 | 0.107 | 0.930 | 0.115 | 0.885 | 0.030 | 0.045 | 0.821 | 0.096 | 0.904 |
| 0.25 | 0.459 | 0.093 | 0.979 | 0.099 | 0.901 | 0.111 | 0.062 | 0.932 | 0.083 | 0.917 |
| 0.20 | 0.828 | 0.085 | 0.996 | 0.086 | 0.914 | 0.493 | 0.069 | 0.991 | 0.071 | 0.929 |
| 0.15 | 1.000 | 0.076 | 1.000 | 0.076 | 0.924 | 0.857 | 0.076 | 0.999 | 0.076 | 0.924 |
| 0.10 | 1.000 | 0.076 | 1.000 | 0.076 | 0.924 | 0.958 | 0.060 | 1.000 | 0.061 | 0.939 |
| 0.09 | 0.994 | 0.077 | 1.000 | 0.077 | 0.923 | 0.964 | 0.061 | 1.000 | 0.061 | 0.939 |
| 0.08 | 0.992 | 0.067 | 1.000 | 0.067 | 0.933 | 0.962 | 0.062 | 1.000 | 0.062 | 0.938 |
| 0.07 | 0.993 | 0.078 | 1.000 | 0.078 | 0.922 | 0.949 | 0.054 | 1.000 | 0.054 | 0.946 |
| 0.06 | 0.991 | 0.081 | 1.000 | 0.081 | 0.919 | 0.954 | 0.057 | 1.000 | 0.057 | 0.943 |
| 0.05 | 1.000 | 0.074 | 0.999 | 0.075 | 0.925 | 0.950 | 0.061 | 1.000 | 0.061 | 0.939 |
| 0.04 | 0.991 | 0.089 | 0.999 | 0.089 | 0.911 | 0.955 | 0.081 | 0.999 | 0.081 | 0.919 |
| 0.03 | 1.000 | 0.071 | 0.999 | 0.072 | 0.928 | 0.962 | 0.068 | 0.999 | 0.068 | 0.932 |
| 0.02 | 1.000 | 0.076 | 0.999 | 0.077 | 0.923 | 0.994 | 0.065 | 1.000 | 0.065 | 0.935 |
| 0.01 | 1.000 | 0.077 | 0.999 | 0.078 | 0.922 | 0.932 | 0.036 | 1.000 | 0.036 | 0.964 |
| 0.00 | 1.000 | 0.042 | 1.000 | 0.042 | 0.958 | 1.000 | 0.042 | 1.000 | 0.042 | 0.958 |

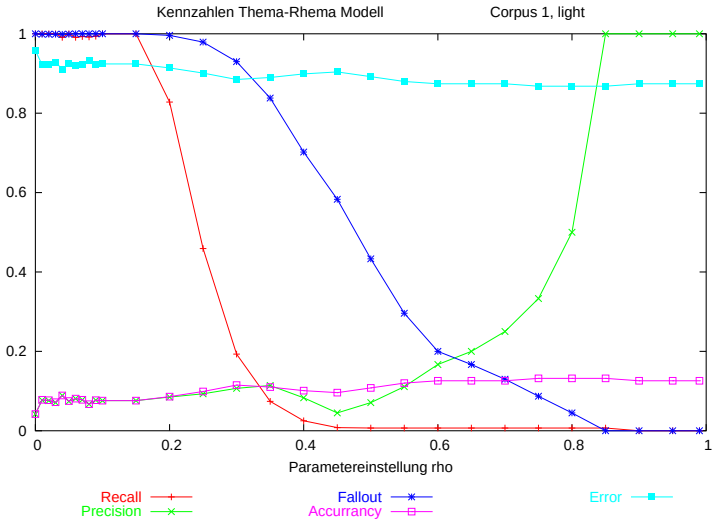



Abbildung 6.9: Th-Rh light, Corpus 1

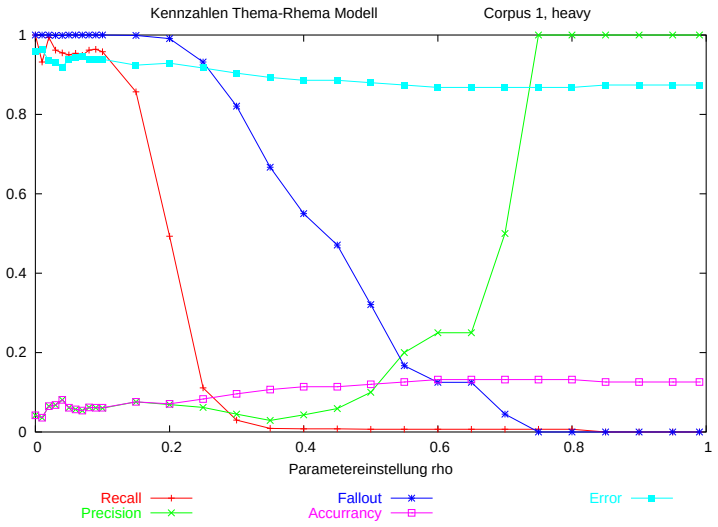



Abbildung 6.10: Th-Rh heavy, Corpus 1

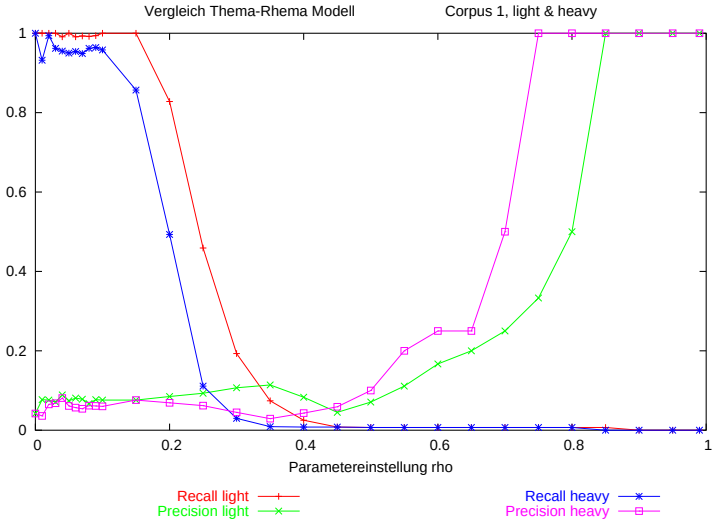



Abbildung 6.11: Th-Rh light vs. heavy, Corpus 1

Beim Einsatz des Thema-Rhema Modells für Corpus 1 (siehe Tabelle 6.8 und Abbildungen 6.9, 6.10 und 6.11) läßt sich ein ähnliches Ergebnis wie schon beim Vektorenmodell feststellen. Die heavy-Variante liegt auch hier ($\rho \leq 0.40$) in den ermittelten Recall- und Precision-Werten hinter denen der light-Variante. Auch der Breakeven Point, der bei der light-Variante 0.16 ($\rho = 0.33$) und bei der heavy-Variante 0.05 ($\rho = 0.29$) beträgt, bestätigt dieses Ergebnis (siehe Tabelle 6.10). Ab Werten von $\rho \geq 0.45$ ist die Precision der heavy-Variante deutlich höher. Bei diesen Werten von $\rho$ liegt jedoch die durchschnittliche Clustergröße bereits bei ca. einem Dokument pro Cluster. Dies läßt darauf schließen, dass der Vergleich zweier Dokumente (1 Testdokument und 1 Clusterzentrum, das einem Dokument entspricht) eine recht gute Retrieval Performanz hinsichtlich der Genauigkeit liefert.

Der interessante Bereich von $\rho$ verläuft bei der light- Variante im Bereich von 0.09, bei der heavy-Variante im Bereich von 0.04. Dies entspricht bei der light-Variante einem Recall/Precision Wertepaar von 0.994/0.077, bei der heavy-Variante von 0.955/0.081.

Tabelle 6.9: Testergebnis Th-Rh, Corpus 2

| $\rho$ | **Corpus 2 - light** | | | | | **Corpus 2 - heavy** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **R** | **P** | **F** | **A** | **E** | **R** | **P** | **F** | **A** | **E** |
| 0.50 | 0.074 | 0.474 | 0.909 | 0.076 | 0.924 | 0.056 | 0.583 | 0.833 | 0.061 | 0.939 |
| 0.45 | 0.080 | 0.321 | 0.950 | 0.076 | 0.924 | 0.056 | 0.538 | 0.857 | 0.061 | 0.939 |
| 0.40 | 0.093 | 0.294 | 0.960 | 0.083 | 0.917 | 0.083 | 0.476 | 0.917 | 0.083 | 0.917 |
| 0.35 | 0.117 | 0.240 | 0.974 | 0.092 | 0.908 | 0.116 | 0.406 | 0.950 | 0.106 | 0.894 |
| 0.30 | 0.242 | 0.237 | 0.987 | 0.141 | 0.859 | 0.185 | 0.258 | 0.980 | 0.127 | 0.873 |
| 0.25 | 0.593 | 0.180 | 0.995 | 0.163 | 0.837 | 0.400 | 0.224 | 0.991 | 0.172 | 0.828 |
| 0.20 | 0.924 | 0.189 | 1.000 | 0.187 | 0.813 | 0.810 | 0.177 | 0.997 | 0.171 | 0.829 |
| 0.15 | 1.000 | 0.205 | 1.000 | 0.205 | 0.795 | 0.948 | 0.129 | 0.999 | 0.129 | 0.871 |
| 0.10 | 1.000 | 0.205 | 1.000 | 0.205 | 0.795 | 0.979 | 0.184 | 0.999 | 0.183 | 0.817 |
| 0.09 | 1.000 | 0.187 | 1.000 | 0.187 | 0.813 | 0.982 | 0.171 | 0.999 | 0.171 | 0.829 |
| 0.08 | 1.000 | 0.152 | 1.000 | 0.152 | 0.848 | 0.983 | 0.182 | 0.999 | 0.182 | 0.818 |
| 0.07 | 1.000 | 0.171 | 1.000 | 0.171 | 0.829 | 0.987 | 0.171 | 0.999 | 0.171 | 0.829 |
| 0.06 | 0.996 | 0.203 | 1.000 | 0.203 | 0.797 | 0.993 | 0.183 | 0.999 | 0.183 | 0.817 |
| 0.05 | 0.995 | 0.208 | 1.000 | 0.208 | 0.792 | 0.982 | 0.191 | 0.999 | 0.191 | 0.809 |
| 0.04 | 0.995 | 0.187 | 1.000 | 0.187 | 0.813 | 0.985 | 0.219 | 0.999 | 0.218 | 0.782 |
| 0.03 | 0.983 | 0.204 | 1.000 | 0.203 | 0.797 | 0.984 | 0.171 | 0.999 | 0.171 | 0.829 |
| 0.02 | 0.995 | 0.180 | 1.000 | 0.180 | 0.820 | 0.979 | 0.180 | 0.999 | 0.180 | 0.820 |
| 0.01 | 1.000 | 0.139 | 1.000 | 0.139 | 0.861 | 0.984 | 0.144 | 0.999 | 0.144 | 0.856 |
| 0.00 | 1.000 | 0.258 | 1.000 | 0.258 | 0.742 | 1.000 | 0.258 | 1.000 | 0.258 | 0.742 |

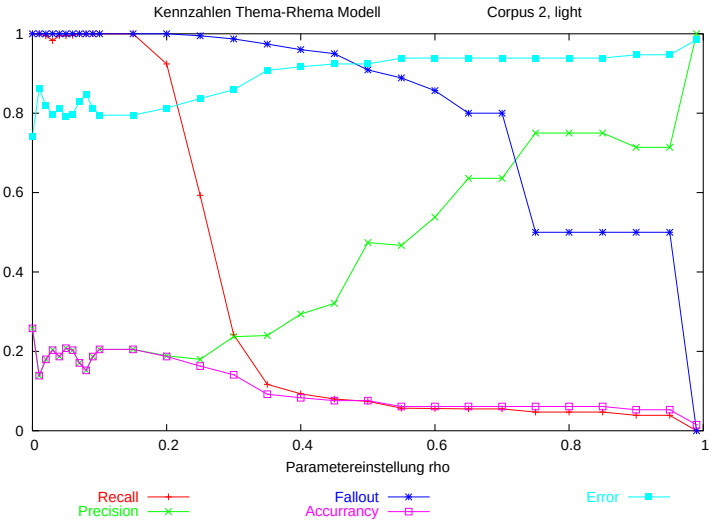



Abbildung 6.12: Th-Rh light, Corpus 2

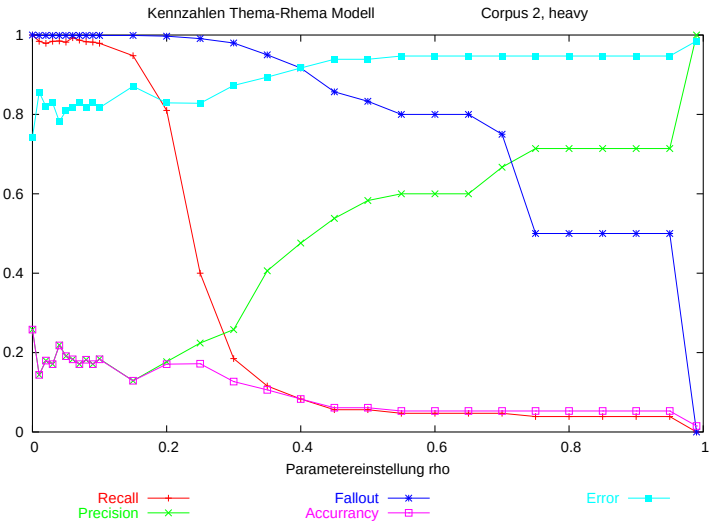



Abbildung 6.13: Th-Rh heavy, Corpus 2

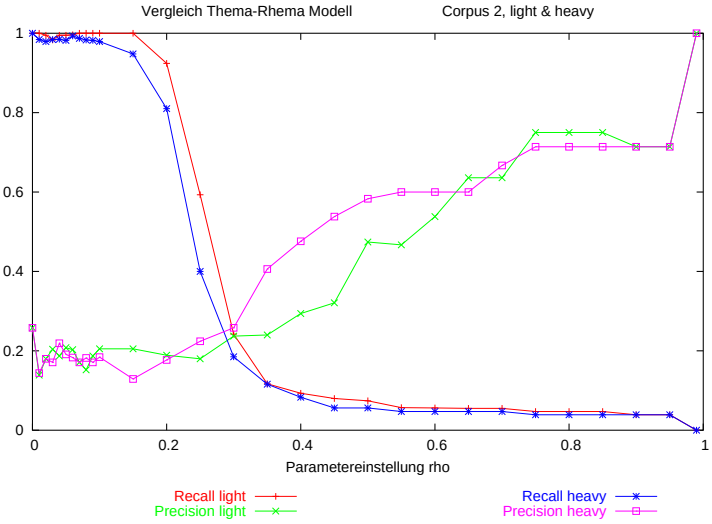



Abbildung 6.14: Th-Rh light vs. heavy, Corpus 2

Bei der Analyse von Corpus 2 mittels des Thema-Rhema Modells (siehe Tabelle 6.9 und Abbildungen 6.12, 6.13 und 6.14) überwiegen die errechneten Kennzahlen wiederum auf der Seite der light-Variante. Der Recall der light-Variante liegt bei allen Werten von $\rho$ über denen der heavy-Variante. Die Precision hingegen erreicht, besonders im Bereich höherer Werte von $\rho \geq 0.45$, bei der heavy-Variante bessere Ergebnisse.

Der Breakeven Point der light-Variante mit 0.24 ($\rho = 0.30$) entspricht nahezu dem Wert der heavy-Variante, der mit 0.25 ($\rho = 0.29$) angegeben ist (siehe Tabelle 6.10).

Der interessante Bereich liegt hier bei Werten von 0.06 für die light-Variante und von 0.04 für die heavy-Variante. Dies entspricht einem Recall / Precision Wert von 0.966 / 0.203 der light-Variante und 0.985 / 0.219 der heavy-Variante.

### 6.4.5 Vergleich: Vektorenmodell vs. Thema-Rhema Modell

Für den direkten Vergleich des Vektorenmodells mit dem Thema-Rhema Modell wurden zusätzlich zu den Kennzahlen Recall, Precision und Breakeven Point das in Kapitel 6.2.2 beschriebene F-Measure hinzugezogen. Eine Gesamtübersicht über die ermittelten Werte geben die Tabellen 6.10 und 6.11. Die Abbildungen 6.15 bis 6.19 spiegeln vergleichende Darstellungen der Ergebnisse wider.

Tabelle 6.10: Testergebnisse - Breakeven Point Analyse

| **Methode** | **Corpus 1** | | **Corpus 2** | |
|---|---|---|---|---|
| | $\rho$ | **BEP** | $\rho$ | **BEP** |
| VSM light | 0.14 | 0.23 | 0.13 | 0.47 |
| VSM heavy | 0.14 | 0.21 | 0.14 | 0.38 |
| Th-Rh light | 0.33 | 0.16 | 0.30 | 0.24 |
| Th-Rh heavy | 0.29 | 0.05 | 0.29 | 0.25 |

Tabelle 6.11: Testergebnisse - F-Measure

| $\rho$ | **Corpus 1** | | | | **Corpus 2** | | | |
|---|---|---|---|---|---|---|---|---|
| | **VSM** | | **Th-Rh** | | **VSM** | | **Th-Rh** | |
| | **light** | **heavy** | **light** | **heavy** | **light** | **heavy** | **light** | **heavy** |
| 0.50 | 0.028 | 0.028 | 0.013 | 0.013 | 0.088 | 0.088 | 0.128 | 0.102 |
| 0.45 | 0.028 | 0.028 | 0.014 | 0.014 | 0.088 | 0.088 | 0.128 | 0.101 |
| 0.40 | 0.028 | 0.028 | 0.038 | 0.013 | 0.088 | 0.088 | 0.141 | 0.141 |
| 0.35 | 0.028 | 0.028 | 0.090 | 0.014 | 0.088 | 0.088 | 0.157 | 0.180 |
| 0.30 | 0.028 | 0.028 | 0.138 | 0.036 | 0.088 | 0.088 | 0.239 | 0.215 |
| 0.25 | 0.027 | 0.027 | 0.155 | 0.080 | 0.088 | 0.088 | 0.276 | 0.287 |
| 0.20 | 0.027 | 0.027 | 0.154 | 0.121 | 0.103 | 0.103 | 0.314 | 0.291 |
| 0.15 | 0.027 | 0.041 | 0.141 | 0.140 | 0.302 | 0.225 | 0.340 | 0.227 |
| 0.10 | 0.186 | 0.159 | 0.141 | 0.113 | 0.383 | 0.361 | 0.340 | 0.310 |
| 0.09 | 0.181 | 0.183 | 0.143 | 0.115 | 0.373 | 0.332 | 0.315 | 0.291 |
| 0.08 | 0.155 | 0.146 | 0.126 | 0.116 | 0.331 | 0.292 | 0.264 | 0.307 |
| 0.07 | 0.126 | 0.134 | 0.145 | 0.102 | 0.299 | 0.295 | 0.292 | 0.291 |
| 0.06 | 0.108 | 0.097 | 0.150 | 0.108 | 0.282 | 0.271 | 0.337 | 0.309 |
| 0.05 | 0.081 | 0.070 | 0.138 | 0.115 | 0.259 | 0.274 | 0.344 | 0.320 |
| 0.04 | 0.082 | 0.075 | 0.163 | 0.149 | 0.236 | 0.217 | 0.344 | 0.358 |
| 0.03 | 0.075 | 0.066 | 0.133 | 0.127 | 0.222 | 0.216 | 0.338 | 0.291 |
| 0.02 | 0.068 | 0.066 | 0.141 | 0.122 | 0.211 | 0.216 | 0.305 | 0.304 |
| 0.01 | 0.081 | 0.069 | 0.143 | 0.069 | 0.259 | 0.274 | 0.244 | 0.251 |
| 0.00 | 0.090 | 0.016 | 0.081 | 0.081 | 0.282 | 0.350 | 0.410 | 0.410 |

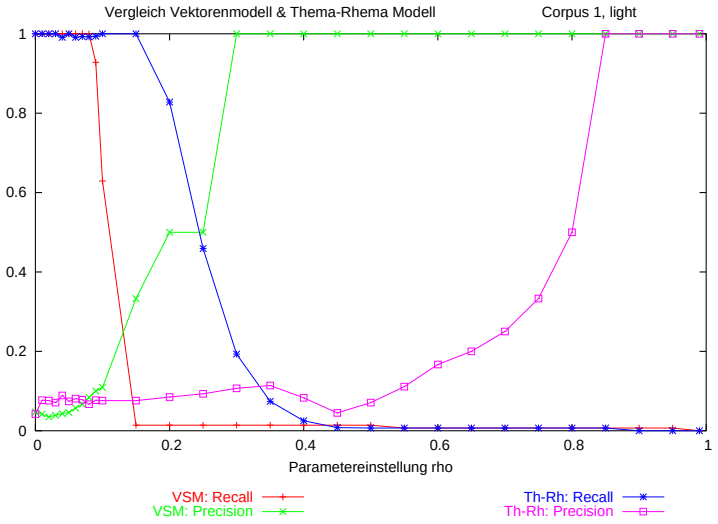



Abbildung 6.15: VSM light vs. Th-Rh light, Corpus 1

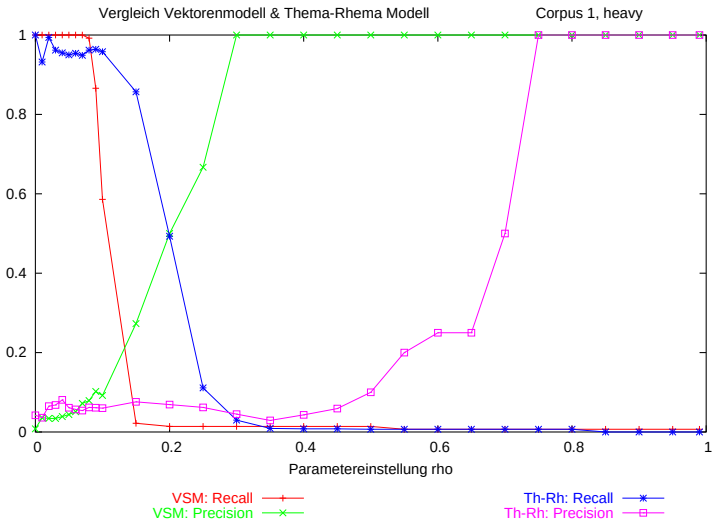



Abbildung 6.16: VSM heavy vs. Th-Rh heavy, Corpus 1

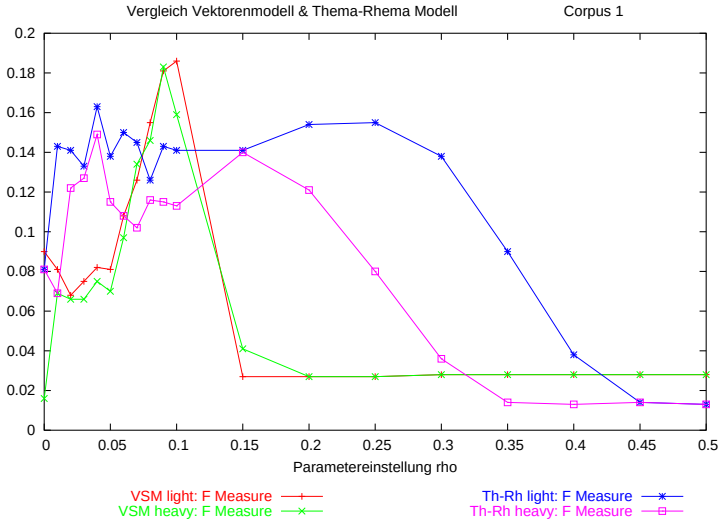



Abbildung 6.17: VSM vs. Th-Rh, F-Measure, Corpus 1

Beim direkten Vergleich des Vektorenmodells mit dem Thema-Rhema Modell läßt sich bei Corpus 1 in beiden Varianten feststellen, dass der Recall beim Thema-Rhema Modell generell besser ausfällt (siehe Abbildungen 6.15 und 6.16). Beim Ermitteln der Precision liefert das Vektorenmodell ab Werten von $\rho \geq 0.07$ deutlich bessere Werte. Im interessanten Bereich beider Modelle liefert das Vektorenmodell bessere Precision-Werte, den besseren Recall erzielt das Thema-Rhema Modell. Dies kehrt sich bei größer werdendem $\rho$ um.

Der Breakeven Point der light-Variante mit 0.23 ($\rho = 0.14$), im Verhältnis zu 0.16 (0.33) der heavy-Variante, liegt bei der light-Variante doch beträchtlich höher. Noch deutlicher zeichnet sich dieses Bild bei den beiden heavy-Varianten ab. Die Breakeven Point Analyse hierbei ergab beim Vektorenmodell einen Wert von 0.21 ($\rho = 0.14$), beim Thema-Rhema Modell einen Wert von 0.05 ($\rho = 0.29$).

Auffallend ist die Rechtsverschiebung bezüglich $\rho$ des Breakeven Points beider Varianten des Thema-Rhema Modells. Dieser Schnittpunkt fällt jeweils in einen Bereich, in dem die Clustergröße annähernd 1 ist. Dies zeigt, dass das Thema-Rhema Modell mit zunehmenden $\rho$ kontinuierlich größere Ähnlichkeiten von Dokumenten und Clusterzentren findet, da es erst ab $\rho \geq 0.50$ zur Bildung einelementiger Cluster kommt. Das Vektorenmodell vollzieht an dieser Stelle eine schärfere Trennung. In nur wenigen Schritten von $0.10 \leq \rho \leq 0.20$ wird jedes Dokument nur einem Cluster zugeordnet. Der Recall des Vektorenmodells erreicht bereits ab $\rho \geq 0.30$ seinen Maximumwert. Anders beim Thema-Rhema Modell, bei dem erst ab Werten von $\rho \geq 0.85$ dieses Ergebnis erreicht wird. Dies läßt sich dadurch begründen, daß das Vektorenmodell bereits bei niedrigeren Werten von $\rho$ zur Bildung einelementiger Cluster führt. Die Spanne vom Zeitpunkt einelementiger Cluster bis zur Erreichung des maximalen Recalls beträgt hier nur zwei Schritte (0.10), beim Thema-Rhema Modell hingegen fünf Schritte (0.25).

Aus dem Vergleich der F-Measure Graphen der beiden Varianten (siehe Abbildung 6.17) ist ebenfalls darauf zu schließen, dass das Vektorenmodell besonders im Bereich von $0.08 \leq \rho \leq 0.12$ bessere Ergebnisse liefert. Die Spitzen der Kurven sowohl der light- als auch der heavy-Variante des Vektorenmodells überragen die Kurven des Thema-Rhema Modells deutlich. Auch die Minimumwerte des Thema-Rhema

Modells liegen unter denen des Vektorenmodells. Jedoch sind die Kurven hier (mit Ausnahme der Spitzen des Vektorenmodell) im Bereich von $\rho \leq 0.4$ konstant höher als die des Vektorenmodells.

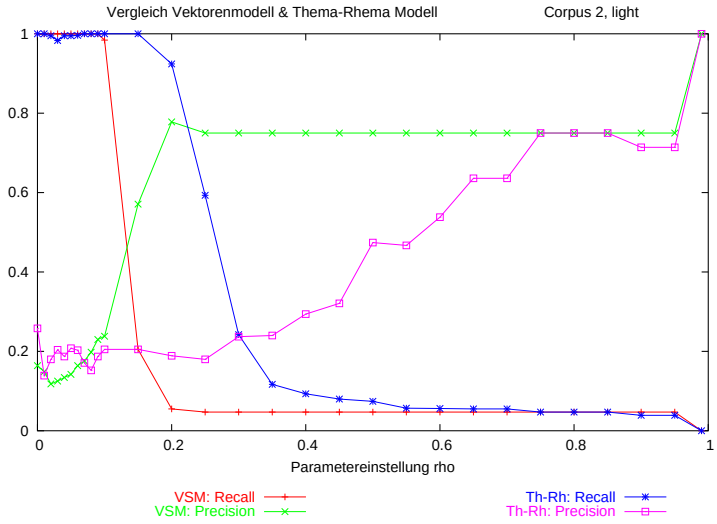



Abbildung 6.18: VSM light vs. Th-Rh light, Corpus 2

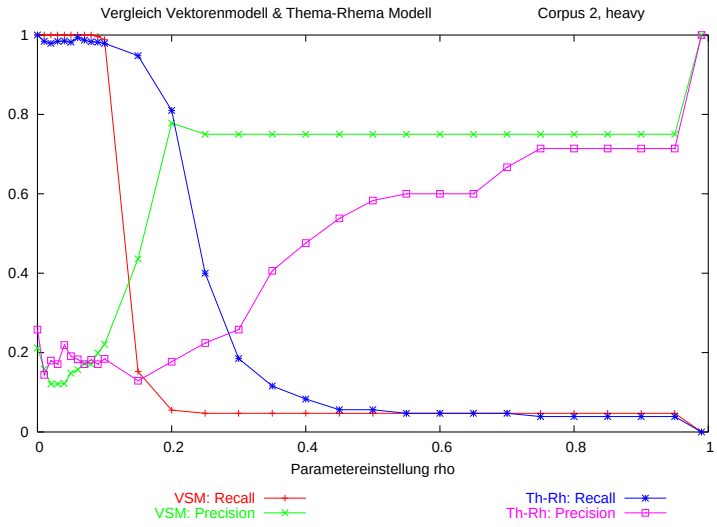



Abbildung 6.19: VSM heavy vs. Th-Rh heavy, Corpus 2

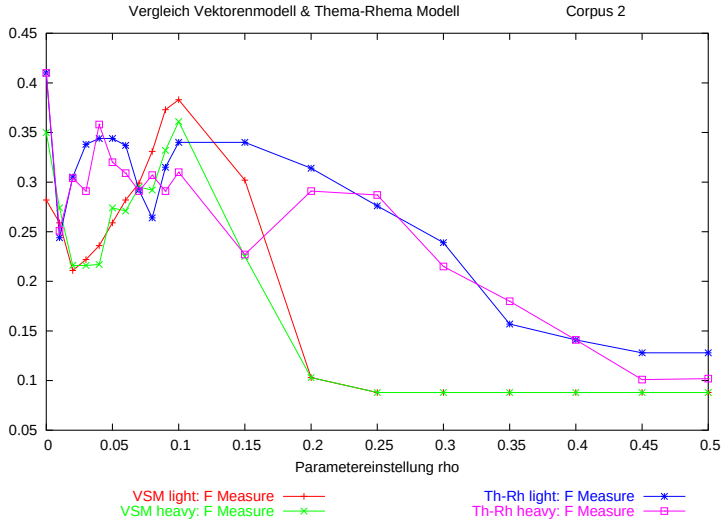



Abbildung 6.20: VSM vs. Th-Rh, F-Measure, Corpus 2

Dasselbe Bild wie zuvor zeichnet sich auch bei Corpus 2 (siehe Abbildungen 6.18 und 6.19) ab. Der Recall liegt beim Thema-Rhema Modell um einiges höher als beim Vektorenmodell, wohingegen die Precision beim Vektorenmodell generell höher verläuft.

Auch die entsprechenden Breakeven Points der light-Varianten, 0.47 ($\rho = 0.13$) des Vektorenmodells und 0.24 ($\rho = 0.30$) des Thema-Rhema Modells, ergeben ein ähnliches Resultat wie die heavy-Varianten (0.38 bei $\rho = 0.14$ für das Vektorenmodell, 0.25 bei $\rho = 0.29$ für das Thema-Rhema Modell). Das Vektorenmodell schneidet besser ab.

Ebenfalls entsprechend ähnlich zum Ergebnis des Corpus 1 fällt die F-Measure Analyse des Corpus 2 aus (siehe Abbildung 6.20). Das Maximum des F-Measure liegt beim Vektorenmodell wiederum höher als beim Thema-Rhema Modell, das Minimum liegt jedoch beim Vektorenmodell niedriger Wiederum erzielt das Thema-Rhema Modell (bis auf die Spitzen des Vektorenmodells) im gesamten Bereich ein besseres Ergebnis hinsichtlich des F-Measure.

Als Endergebnis kann festgestellt werden, dass das Vektorenmodell bessere Spitzenergebnisse in beiden Corpora und in beiden Varianten liefert. Sowohl der Breakeven Point als auch das F-Measure bestätigen dies. Gesamt gesehen liegt die Precision ebenfalls beim Vektorenmodell höher. Im Gegensatz dazu fällt die Ermittlung des Recalls besser für das Thema-Rhema Modell aus.

Jener Punkt, an dem durschnittlich nur ein Dokument pro Cluster gebildet wird, ist besonder wichtig. In diesem Fall findet kein eigentliches Clustering statt, und das Verfahren ist mit traditionellem Information Retrieval gleichzusetzen. Dieser Punkt wird durch das Vektorenmodell ($\rho \geq 0.20$) um einiges früher erreicht als beim Thema-Rhema Modell ($\rho \geq 0.50$).

Das Miteinbeziehen von zusätzlichen Wortkategorien wie Adjektiven und Adverbien der heavy-Varianten der einzelnen Modelle wirkte sich nur geringfügig auf das erzielte Ergebnis aus, führt jedoch zu Umfangseinschränkungen der verwendeten Trainingscorpora (durch Listen- und Feldbegrenzungen im Speicher) sowie zu beträchlich längeren Verarbeitungszeiten. Dieses Ergebnis fällt unerwartet aus, da zusätzliches Material bei der Bildung der Dokumentrepräsentationen eher als hilfreich gesehen

wurde.

### 6.4.6 Gewonnene Erkenntnisse

Im Zuge der Evaluation wurden mittels zweier Gewichtungsmodelle, dem Vektorenmodell und dem Thema-Rhema Modell, zwei unterschiedliche Corpora ausgewertet. In beiden Modellen wurden jeweils zwei Varianten, welche auf unterschiedlichen Wortarten aufbauten, verwendet.

Die zwei Gewichtungsverfahren wurden hierbei verwendet, um den Beitrag der Thema-Rhema Theorie zur Textanalyse festzustellen. Die jeweiligen light- und heavy-Varianten dieser Modelle wurden eingesetzt, um den Beitrag von Adjektiven und Adverbien zur Dokumentrepräsentation aufzuzeigen. Der Einsatz zweier Corpora galt der Überprüfung des Ergebnisses anhand empirischer Erfahrungen sowie einer möglichst objektiven Abschätzung der erzielten Ergebnisse.

Der Vergleich der zwei Gewichtungsmodelle ergab eine graduellere Clusterbildung des Thema-Rhema Modells, während das Vektorenmodell grundsätzlich kleinere Cluster bildete. Die errechneten Kennzahlen des Vektorenmodells liegen hierbei über denen des Thema-Rhema Modells. Das Thema-Rhema Modell führt durch das Rhema-Thema Mapping implizit eine Reduktion des Vokabulars durch, da bei der Repräsentation nur Themata verwendet werden.

Die Berücksichtigung von Adjektiven und Adverbien hingegen ergab keine besonderen Änderungen der Resultate, führte jedoch zu einem Anwachsen der zu Verarbeitenden Datenmenge (ein Drittel mehr Indexterme). Dadurch bildeten sich in den jeweiligen heavy-Varianten durchschnittlich größere Cluster. Durch den Einsatz der zwei Corpora ergab sich ebenfalls ein nur geringer Unterschied der jeweiligen light- und heavy-Varianten.

Die Kennzahlenkurven der Varianten der Gewichtungsmodelle waren in beiden Corpora einander sehr ähnlich. Der eingeschränktere Wortschatz und die höhere Anzahl von Dokumenten pro Kategorie des spezielleren Corpus 2 wirkte sich bis auf die Berechnungszeiten nicht sonderlich auf die Ergebnisse aus. Das Vektorenmodell erreichte jedoch im direkten Vergleich mit dem Thema-Rhema Modell in beiden

Corpora ein etwas besseres Gesamtergebniss. Die Breakeven Point Analyse ergab in beiden Corpora in beide Modellen das beste Ergebnis in Bereichen, in denen ein Dokument einem Cluster entspricht. Auch hier lieferte das Vektorenmodell bessere Ergebnisse. In Bezug auf den Recall hingegen schlug das Thema-Rhema Modell das Vektorenmodell.

## 6.5 Dokumentclustering und Information Retrieval

Textclustering kann im Bereich des Information Retrievals eingesetzt werden, um den Suchraum für Abfragen an eine große Datenmenge in annehmbaren Antwortzeiten einzuschränken. Das Grundprinzip des Clusterretrievals wurde bereits im Kapitel 2.5 vorgestellt.

Eine Datensammlung wird anhand eines Clusterverfahrens in einzelne, sich nicht überlappende Teilmengen aufgespalten. Dies geschieht ohne jegliche Interaktion mit einem Benutzer und findet als Batchverarbeitung statt. Sobald die Cluster gebildet sind, ist das System einsatzbereit. Bei einer eingehenden Abfrage wird jedes Clusterzentrum mit dieser Abfrage verglichen, wobei ein Ähnlichkeits- oder Übereinstimmungswert berechnet wird. Nach der Bewertung aller Cluster auf diese Art werden nur diejenigen Cluster in Betracht gezogen, deren Ähnlichkeitswert über einer festgelegten Grenze ist. In SyRS stellt dies der Parameter $\rho$ dar. Anschließend werden alle Dokumente der übrigen Cluster an den Benutzer zurückgeliefert, wobei diese nach Übereinstimmungsgrad ihrer Cluster sortiert werden. Eine endgültige Auswahl oder verfeinerte Suche findet manuell vom Benutzer statt.

Dazu folgendes Beispiel: Während des Trainings mit Corpus 2 mit der heavy-Variante des Thema-Rhema Modells wurden 207 Dokumente auf 167 Cluster aufgeteilt ($\rho = 0.20$). Unter den Trainingsdokumenten befanden sich zufällig auch die 5 Dokumente c107_1610.res, c107_1851.res, c107_424.res, c107_774.res, c107_972.res. Das Kürzel „c107“ steht dabei für die Kategorie „Betriebswirtschaft - Funktional / Organisation / Gruppen, Teamarbeit“. Die Dokumentrepräsentationen floßen während des Trainierens in die Bildung der Clusterzentren mit ein.

Nachdem das Netz fertig trainiert wurde, konnte es für das eigentliche IR eingesetzt werden. Während der Testphase wurde das System mit einem neuen, unbekannten Text konfrontiert, der dieselben Analyseschritte wie die Trainingsdokumente zuvor durchging. Dieser Text stammte aus Kontrollzwecken aus derselben Kategorie „c107“ wie die 5 Dokumente, von denen man wußte, das sie in der Trainingsmenge waren. Das System verglich während dieser Phase jedes der 167 Clusterzentren mit dem neuen Text. Als Ergebnis wurden die Cluster 23, 50 und 160, die folgende Texte enthielten, als ähnliche Cluster zum Testdokument identifiziert:

Tabelle 6.12: Beispiel eines Abfrageergebnisses (1)

| **Cluster** | **Dokumente** | **Übereinstimmung** |
|---|---|---|
| 23 | c107_424.res | 0.259 |
| 50 | c218_2086.res, c262_5544.res | 0.208 |
| 160 | c107_4451.res, c107_5016.res | 0.228 |

Es wurden somit 3 der 5 vorhandenen Dokumente in den 3 als relevant ermittelten Clustern gefunden. Alle Dokumente dieser Cluster wurden in weiterer Folge an den Benutzer zurückgeliefert. Nach einer zusätzlichen Sortierung der Dokumente nach den Übereinstimmungsgraden ihrer Cluster resultierte der Retrievalvorgang in folgender Weise:

Tabelle 6.13: Beispiel eines Abfrageergebnisses (2)

| **Dokument (Cluster)** | **Übereinstimmung** |
|---|---|
| c107_424.res (23) | 0.259 |
| c107_4451.res (160) | 0.228 |
| c107_5016.res (160) | 0.228 |
| c218_2086.res (50) | 0.208 |
| c262_5544.res (50) | 0.208 |

Eine zusätzliche, cluster-interne Reihung wurde hierbei nicht durchgeführt. Dies kann aber durch weitere Vergleiche der einzelnen Dokumente mit ihrem Cluster-

zentrum inkludiert werden. Eine größere Ähnlichkeit eines Dokuments mit seinem Clusterzentrum entspräche dann einer höheren Reihung.

Der Benutzer kann anhand eines solchen Ergebnisses die wirklich gewollten Dokumente aus der Liste entweder selbst auswählen (im Falle weniger Dokumente), oder auch ein weiteres Suchverfahren anwenden, um diese eingeschränkten Resultatmenge noch etwas genauer auf seine Inhalte hin einschränken.

KAPITEL 7

# Resümee

## 7.1 Zusammenfassung

Ihren Ausgangspunkt nimmt diese Arbeit an dem zuhnemhmend an Bedeutung erlangenden Thema des Information Retrieval. Hierbei wird von großen, unstrukturierten und heterogenen Datenmengen ausgegangen, die mittels automatischer Verfahren organisiert, verwaltet und gezielt bezogen werden können. Meist handelt es sich hierbei um natürlichsprachliche Texte, weshalb diese Arbeit auf deutschsprachigen Textdokumenten als zugrundeliegenden Daten aufbaut. Ein Teilgebiet des Information Retrievals ist die Textgruppierung. Sie verwirklicht das Prinzip des Aufteilens der gesamten Datenmenge in überschaubare Teile, wobei ähnliche Textdokumente in denselben Gruppen zum Liegen kommen. Eine solche Vorabgliederung erleichtert sowohl die Organisation als auch das Wiederauffinden von gewünschten Informationen.

Die Arbeit selbst gliedert sich in zwei Teile:

Die ersten Kapitel 2, 3 und 4 behandeln den theoretischen Hintergrund von Textgruppierungen. Hier werden allgemeine Begriffsabgrenzungen vorgenommen und einführende Grundlagen in das Information Retrieval erläutert. Dazu werden verschiedene Aufgabengebiete wie das Textretrieval, die Textfilterung und die Textgruppierung genauer betrachtet. Ein wichtiger Punkt, an dem vorhandene Systeme oftmals scheitern, ist das Finden einer geeigneten Repräsentationsform zur Darstellung der Informationen von Texten. Der Umfang der Datenmenge stellt dabei nicht die einzige Schwierigkeit dar. Auch syntaktische Freiheiten und semantische Ambiguitäten machen es schwierig, geeignete Repräsentationen automatisch abzuleiten.

Bei näherer Betrachtung von Textgruppierungen kann diese in zwei Teilgebiete auf-

gespalten werden. Die Dokumentkategorisierung beschäftigt sich mit Gruppierungen anhand von (meist menschlichen) Beispielskategorisierungen. Die Kategorien sind hierbei bereits durch einen Namen festgelegt und vor dem eigentlichen Gruppierungsprozess bekannt. Beim Dokumentclustering hingegen sind lediglich die zu gruppierenden Dokumente ohne jegliche Zusatzinformation vorhanden. Die Kategoriedefinitionen müssen somit selbst aus dem Gesamtumfang der Dokumente generiert werden. Um Übereinstimmungen von Dokumenten und Dokumentgruppen finden zu können, werden Ähnlichkeits- und Abstandsmaße benutzt. Diese Maße werden von allen Modellen zur Gruppierung von Dokumenten verwendet, seien es nun statistische Modelle, probabilistische Modelle oder Ansätze aus dem Bereich des Machine Learnings.

Im Zusammenhang mit Textgruppierungen ist besonders die Kombination mit Techniken aus dem Bereich des maschinellen Lernens interessant. Oft werden Genetische Algorithmen eingesetzt, um eine Textgruppierung, die auch als Optimierungsproblem formuliert werden kann, durchzuführen. Einen andereren und besonders attraktiven Zugang zu dieser Problematik stellen Neuronale Netze dar. In der hier vorliegenden Arbeit kamen zwei unterschiedliche Typen Neuronaler Netze, ein Fuzzy Associative Memory Netz (FAM) und ein fuzzy Adaptive Resonance Theory Netz (ART), zum Einsatz. Das FAM übernahm hierbei die Aufgaben der Termgewichtung, während das fuzzy ART Netz das eigentliche Clustering durchführte.

Wie schon zuvor angesprochen, spielt die Dokumentrepräsentation eine zentrale Rolle bei automatischen Textgruppierungssystemen. Ist eine gewählte Repräsentation zu ungenau, kann das System keine guten Ergebnisse liefern. Ist sie jedoch zu detailliert, kann dies zu endlosen Berechnungen mit unrealistischen Antwortzeiten führen. Deshalb sind verschiedene Methoden der Textanalyse notwendig, um relevante Informationen in Texten zu identifizieren und so deren Inhalte sinngemäß wiederzugeben. Für diese Aufgabe kommen Methoden wie die Tokenisierung (Aufbereitung), das Tagging (Vergabe von Wortkategorien), das Stemming (Stammformbildung) und die Filterung (durch Stopwortlisten) zum Einsatz. Anschließend an diese Textanalyse werden die im Text gefundenen Indexterme in Listenform abgelegt. Zusätzlich findet noch eine Gewichtung der Indexterme statt. Ein Standardmodell hierfür stellt

das Vektorenmodell dar. Einzelne Begriffe werden in diesem Modell unter Berücksichtigung der Auftrittshäufigkeit im Text und der Auftrittshäufigkeiten in anderen Texten entsprechend bewertet. Beziehungen von Wörtern untereinander werden bei diesem Ansatz nicht miteinbezogen. Ein anderes Vorgehen findet bei der Gewichtung mittels des Thema-Rhema Modells statt. Hier werden einzelne Sätze in zwei Teile geteilt: Ein erster Teil, das Thema, beinhaltet die Grundaussage des Satzes, ein zweiter Teil, das Rhema, ergänzt und erklärt den ersten Teil. Zur Repräsentation eines Textes wird nur der thematische Teil verwendet. Zusätzlich zur Auftrittshäufigkeit wird bei der Gewichtung jedoch auch der Beitrag des rhematischen Materials zur Bildung von thematischen Konzepten miteinberechnet. Somit berücksichtigt dieses Modell ebenfalls Beziehungen zwischen Wörtern.

Der zweite große Block dieser Arbeit, der die Kapitel 5 und 6 umfasst, ist praktischer Natur. Hier wird ein Prototyp für den Einsatz des Dokumentclusterings vorgestellt und im Detail beschrieben. SyRS (Systemic Retrieval System) besteht aus zwei Hauptmoduln, dem Natural Language Modul und dem Neuronal Netzwerk Modul.

Das Natural Language Modul übernimmt die gesamten Aufgaben der Textanalyse und der Gewichtung. Hier kommen die zuvor beschriebenen Methoden der Tokenisierung, des Taggings, des Stemmings, und der Filterung zum Einsatz. Die Gewichtung geschieht anhand der zwei vorgestellten Modelle, des Vektorenmodells und des Thema-Rhema Modells. Die ermittelten Ergebnisse werden an das Neuronal Netzwerk Modul weitergegeben. Zusätzlich zu den zwei Gewichtungsmodellen werden auch verschiedene Wortgruppen bei der Repräsentationsbildung verwendet.

Das Neuronal Netzwerk Modul stellt die lernende Komponente des Systems dar. Es besteht aus zwei Neuronalen Netzen, einem Fuzzy Associative Memory (FAM) und einem fuzzy Adaptive Resonance Theory Netz (ART). Das FAM wirde nur in Verbindung mit dem Thema-Rhema Modell verwendet. Es berechnet den Beitrag rhematischen Materials bei der Gewichtung der thematischen Indexterme. Das ART hingegen führt das eigentliche Clustering durch. Beide Netze bauen bei ihren Berechnungen auf zuvor erlernten Trainingsbeispielen auf.

Um die Ergebnisse eines solchen Clusteringprozesses zu beurteilen und diese mit anderen vergleichen zu können, müssen geeignete Metriken und Testdaten vorhanden

sein. Als gängige Kennzahlen hierfür werden Recall und Precision, der Breakeven Point oder Kombinationen aus Recall und Precision, wie das F-Measure, vorgestellt. Diese drücken die Eigenschaften, wie beispielsweise die Genauigkeit oder die Vollständigkeit eines Retrievalverfahrens, in Zahlenwerten aus. Ebenfalls existieren verschiedenste Testdatensammlungen (Corpora) zur Auswertung solcher Systeme. Leider sind diese fast ausnahmslos für das Englische zu finden. Deshalb wird im Laufe dieser Arbeit ein auf der Diplom- und Dissertationsdatenbank Diplomica.com basierendes Textcorpus entwickelt. Anhand derselben Testdokumente und derselben Kennzahlen werden die zwei Gewichtungsmethoden, das Vektorenmodell und das Thema-Rhema Modell, in einer Vielzahl von Tests miteinander verglichen und kritisch beleuchtet. Ziel dieser Arbeit wist eine objektive Bewertung und Gegenüberstellung dieser Verfahren.

Als Ergebnis der Untersuchungen ist festgestellt worden, dass eine Gewichtung über das Thema-Rhema Modell ähnliche Werte wie die des Standard-Vektorenmodells liefert. Keines der beiden Modelle überragt bei der Auswertung das Andere deutlich. Durch die verschiedenen Parametereinstellung kann die Clusterbildung klar nachvollzogen werden. Grundsätzlich sind die Cluster sehr klein und entsprechen oft keiner der Vergleichskategorien, weshalb die erzielten Kennzahlenergebnisse als pessimistisch bezeichnet werden können. Die durchschnittliche Clustergröße erreicht erst bei sehr kleinen Ähnlichkeitsübereinstimmungen ($\rho$) den entsprechenden Wert der ursprünglichen Kategorisierung.

Die Vergleiche werden aufgrund der Kennzahlen Recall, Precision, dem Breakeven Point und dem F-Measure durchgeführt. Im Bereich der ähnlichen Durchschnittsclustergröße (geringe $\rho$-Werte) sind die Precision und das F-Measure verhältnismäßig niedrig, der Recall hingegen erreicht sein Maximum. Der Bereich des Breakeven Points, an dem Recall und Precision denselben Wert annehmen, liegt beim Vektorenmodell höher als beim Thema-Rhema Modell. Jedoch liegt dieser in beiden Modellen bereits in einem Bereich, in dem nur einelementige Cluster existieren.

Auch eine genauere Textanalyse, bei der neben Nomen und Verben auch Adjektive und Adverbien miteinbezogen wurden (den heavy-Varianten), führt in beiden Modellen nur zu unwesentlichen Änderungen des Ergebnisses. Einerseits kann ei-

ne genauere Untersuchung der vorhandenen Parametereinstellungen der Neuronalen Netze zu anderen Ergebnissen führen. Andererseits lässt das erzielte Resultat darauf schließen, dass der Einsatz geeigneterer Repräsentationen von deutschen Texten ebenfalls sinnvoll und notwendig ist. Hierzu müssten die Methoden der Textanalyse verfeinert und ergänzt werden. Im nächsten Abschnitt werden einige dieser Ansätze vorgestellt.

Eines der großen Probleme des Information Retrieval ist sicherlich auch der Umfang der Datenmenge. Die Einschränkung der Anzahl unterschiedlicher Indexterme führt jedoch in weiterer Folge auf eine geringe Anzahl von Corpusdokumenten. Schon durch diese extreme Reduktion des Wortschatzes (252 und 211 Dokumente) dauert die Trainingsphase für eine Parametereinstellung bis zu 20 Stunden auf einem Pentium 4 mit 1700 MHz und 256 MB Hauptspeicher. Dies zeigt den Bedarf optimierter Berechnungsverfahren und Techniken, ohne die gute Ergebnisse nur schwer erzielt werden können.

## 7.2 Erweiterungspotential von SyRS

### 7.2.1 Das Natural Language Modul

Der Tokenizer führt sowohl eine Textformatierung als auch eine Textersetzung durch. Besonders im Bereich der Abkürzungsersetzung und Formaterkennung (z.B. Datumsformate, Kreditkartennummern, e-Mail Adressen, ...) sind hier noch viele Ergänzungen möglich und sinnvoll. Ebenso können auftretende Sonderzeichen (z.B. Bindestriche) vorverarbeitet beziehungsweise ausgefiltert werden. Da der in SyRS verwendete Tokenizer in Perl entwickelt wurde, können dieser Ergänzungen relativ einfach umgesetzt werden. Auch eine Portierung in andere Programmiersprachen, die bereits fertige Funktionsbibliotheken für eine Textverarbeitung zur Verfügung stellen, ist sicherlich sinnvoll. Wichtig ist es, das Regelwerk des Tokenizers von der Implementierung getrennt zu halten (z.B. in Form von textbasierten Regelwerken), um auch in Zukunft weitere Anpassungen vornehmen zu können.

Der Tagger arbeitet aufgrund seines rein probablistischen Verfahrens sehr effizient.

Dennoch sind Fehlkategorisierungen von Wörtern ein oft auftretendes Problem, da eine bereits vortrainierte Version zum Einsatz kam, die einen Zugriff auf das Regelwerk nicht zulässt. Eine alternative Möglichkeit stellt der brill_Tagger dar. Aufgrund seiner inkrementellen Erweiterungsmöglichkeiten könnte er bessere Ergebnisse liefern. Da der Tagger jedoch nicht für das Deutsche in trainierter Form vorliegt (außer als Forschungsprojekt an der Universität Zürich), müsste hier einiges an Zeit und Arbeit investiert werden, um ihn selbst zu Trainieren. Doch die Anpassung des brill_Taggers bringt sicher große Vorteile mit sich, wie z.B. ein erweiterbares Lexikon und somit die Möglichkeit domänenspezifischen Wissens und eine uneingeschränkte Erweiterung des verwendeten Regelwerks. Da der brill_Tagger im native C-Code vorliegt, sollte auch eine Erweiterung um Wortlemmata, wie sie der tree_Tagger verwendet, in den Grenzen des Möglichen liegen.

Obwohl der vorliegende Stemmer arbeitet sehr effizient, macht er dennoch einige Wortformenrückführungen undurchsichtig und teilweise falsch. Dieser Effekt wird durch den Einsatz des gleichen Stemmers beim Lernen als auch beim Testen der Texte verringert. Als interessante Verbesserung sei hier ein morphologischer Parser vorgeschlagen, der jedem Wort eine syntaktische Struktur zuweist. Solche Parser werden oft in Prolog implementiert, wobei sie während der Wortanalyse Baumstrukturen aufbauen. Beispielsweise könnte ein Wort wie „angelacht“ auf die Komponenten „an-ge-lach-t“ zurückgeführt werden. Der Wortstamm wäre in diesem Fall „lach“, „ge“ ist ein Zeitoperator (Partizip), „t“ gibt Eigenschaften wie die Peson oder die Zeit an, und „an“ ist ein angefügter Verb-Partikel. Ebenfalls könnte die Problematik der Nominalkomposita auf diese Art behandelt werden. Wörter wie „Hosenknopf“ oder „Dichtungsring“ könnten somit in ihre Komponenten aufgebrochen werden zu „Hosen-Knopf“ und Dichtung-s-Ring. Dies könnte zu besseren Ergebnissen der Dokumentrepräsentationen und Ähnlichkeitsbestimmungen führen. Mithilfe eines morphosyntaktischen Regelwerks könnte somit ein eigenes Lexikon entwickelt werden, wobei Wortformenrückführungen einfach und korrekt durchgeführt werden können. Wie schon zuvor ist es auch hier wichtig, die einzelnen Regeln durchsichtig und vom Code getrennt zu halten. Aufgrund des großen Regelapparats ist es jedoch fraglich, wie effizient ein solcher Parser arbeiten kann.

Oftmals werden in Texten verschiedene Wörter verwendet, um ohne oftmalige Wortwiederholungen denselben Sachverhalt auszudrücken. Eine nicht unbeträchtliche Menge der identifizierten Indexterme sind deshalb sinngemäß gleich (synonym), entgegengesetzt (antonym) oder gleichlautend (homonym). Eine Verbesserung des Systems könnte über einen Einbau solcher Lexika, die diese Beziehungen beinhalten, durchgeführt werden. Dadurch kann sowohl die Ausdrucksstärke von Indextermen durch eine bessere Konzeptbildung als auch die Effektivität und Effizienz durch die weitere Reduktion von Termen (im Fall von Synonymen) gesteigert werden. Als eines der bekanntesten Lexika für diese Aufgabenstellung ist sicherlich WordNet zu nennen.

Die bei der Textanalyse und -gewichtung eingesetzte Thema-Rhema Theorie brachte nur unwesentliche Erfolge. Das Aufgabenspektrum des Modells beschränkt sich auf die Identifizierung semantisch gehaltvoller Wörter und deren Gewichtung. Die Algorithmen zur Findung dieser Wörter sind jederzeit erweiterbar und so für spezielle Domänen anzupassen. Eine Erweiterung dieser Termidentifikationsregeln, wie etwa die Thema-Rhema Bildung von aufeinanderfolgenden Nomen, scheint jedoch zu keiner Verbesserung des Ergebnisses zu führen. Der Einsatz eines Natural Language Parsers könnte an dieser Stelle eine tiefere, syntaktische und semantische Analyse durchführen. Eine genauere Erfassung der Syntax ermöglicht ein genaueres Identifizieren von bedeutenden Informationen, wodurch auch pronominale Ersetzungen aufgrund erkannter semantischer Merkmale wie Thetarollen durchgeführt werden könnten. Ebenfalls wäre eine Analyse komplexer Relativsatzkonstruktionen umsetzbar, die ohne jegliche linguistische Theorie nur sehr schwer möglich ist. Auch komplizierte Nominalphrasenschachtelungen könnten dadurch richtig aufgelöst werden. Ein denkbarer Ansatz hierzu wäre die Verwendung eines auf Prolog basierenden NTMS-Parsing-Systems, wie es Dr. Günther Fliedl im Zuge seiner Habilitation [23] und des NIBA-Projekts [1] an der Universität Klagenfurt entwickelt. Wie schon beim morphologischen Parser angesprochen stellt sich auch hier die Frage der Performanz.

[1] http://www.ifi.uni-klu.ac.at/IWAS/HM/Projects/NIBA/OELT2001Fliedl.pdf
(Stand: 2003.10.18)

### 7.2.2 Das Neuronalen Netzwerk Modul

Die in SyRS verwendeten Neuronalen Netze liefern für eine Reihe Themata und Rhemata (Dokumentrepräsentationen) als Input bestimmte Clusterzuweisungen zurück. Einerseits wird in einem ersten Schritt der Einfluss der Rhemata auf die Gewichtung der Themata durch ein Fuzzy Associative Memory Netz berechnet. Anschließend werden die gewichteten Themata einzelnen Textclustern zugeordnet.

Durch die Überführung von rhematischem auf thematisches Material findet eine Art Konzeptbildung statt. Solche Konzeptbildungen scheinen bei der Aufgabe einer Generalisierung des Dokumentinhaltes und dessen Vergleich mit anderen Dokumenten von Nutzen zu sein. Da dies in SyRS durch ein selbstorganisierendes Neuronales Netz vollautomatisch geschieht, stellt sich die Frage, ob dies nicht auch durch den zusätzlichen Einsatz eines Regelwerks unterstützt werden könnte. Da das Neuronale Netz alleine keinerlei Aussage darüber zulässt, wie das berechnete Ergebnis zustande gekommen ist, könnte eine Kombination aus einem Regelwerk und einer lernenden Komponente zu einer Verbesserung der Konzeptbildungen führen.

Der Clustering-Algorithmus läuft in der Lernphase per Definition anders ab als in der Testphase. Dadurch nimmt SyRS eine reine Clusterung der Dokumente vor, bei der jedes Dokument während der Lernphase genau einem Cluster zugewiesen wird. Beim Testen hingegen werden alle Cluster berücksichtigt, die einen gewissen Grenzwert an Ähnlichkeit überschreiten. Unter Umständen wäre eine Überlegung von Mehrfachzuweisungen während der Lernphase ebenfalls sinnvoll. Das dieses Vorhaben jedoch sehr rechenintensiv und mathematisch komplex ist, bleibt es weiteren Nachforschungen vorbehalten. Zusätzlich ist das Clusteringergebnis während des Lernens von der Reihenfolge der Inputdokumente abhängig. Dies kann durch ein oftmaliges Trainieren des Neuronalen Netzes mit denselben Daten (solange bis das Ergebnis stabil bleibt) verhindert werden. Durch die oben genannten Mehrfachzuweisungen (mittels entsprechenden Gewichtungen) könnte dieser Effekt abgeschwächt werden.

Eine weitere nützliche Eigenschaft wäre das Zusammenlegen von verschiedenen Clustern. Werden beispielsweise zwanzig Cluster gebildet, kann es vorkommen, dass zwei Cluster Dokumente derselben Kategorie enthalten. In einem solchen Fall wäre es wünschenswert, diese zwei Cluster miteinander Vereinen zu können, wodurch in der

Testphase wiederum ein Vergleich der Abfrage mit einem Clusterzentrum eingespart werden könnte. Ein Anhaltspunkt für solche Cluster, die zusammengelegt werden könnten, wäre beispielsweise ein annähernd gleicher Ähnlichkeitswert beim Vergleich der Abfrage mit den Clusterzentren. Umgesetzt könnte dies entweder durch ein einfaches Regelwerk oder mittels hierarchischen Clusteringmethoden werden. Entsprechen zwei Cluster $x$ und $y$ derselben Kategorie, könnte bei einer Übereinstimmung einer Abfrage mit Cluster $x$ automatisch auch Cluster $y$ (und umgekehrt) bezogen werden, ohne dass ein zusätzlicher Vergleich mit Cluster $y$ stattfindet. Eine weitere Möglichkeit ist ein hierarchischen Clusteringverfahren (siehe Kapitel 2.5) zu verwenden, bei der Cluster zu Meta-Clustern zusammengefaßt werden. Somit erhält man Cluster, die wiederum aus mehreren Clustern bestehen.

Ein erster Verbesserungsvorschlag könnte wie in Abbildung 7.1 aussehen.

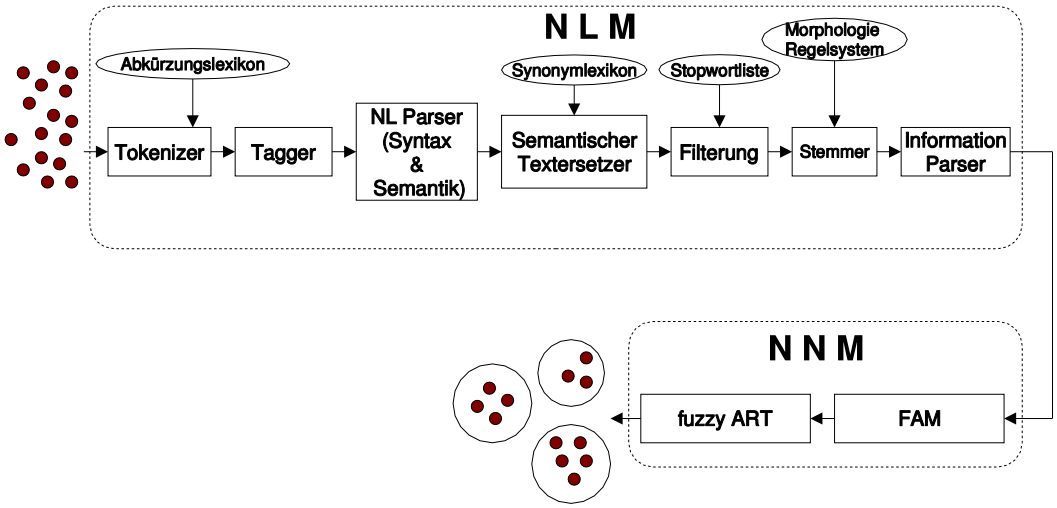



Abbildung 7.1:

Der Tokenizer ist hier um ein Abkürzungslexikon ergänzt, welches die gängigsten Abkürzungen enthält und im Text ersetzt. Ein Natural Language Parser liefert (zusätzlich zu den Wortkategorien des Taggers) die syntaktische Struktur der Sätze des Textes sowie grundlegende semantische Informationen wie den Akteur einer Handlung. Dadurch kann der semantische Textersetzer nicht nur Synonyme aufgrund des Synonymlexikons aufschlüsseln, sondern beispielsweise auch Pronominalersetzungen durchführen.

## 7.3 Ausblick

In dieser Arbeit wurde versucht auf, die Probleme des Information Retrievals, im Besonderen die der Textgruppierung, einzugehen. Der erweiterte SyRS Prototyp soll als mögliches Beispiel und Anstoß für eine Lösung der unzähligen Schwierigkeiten dieses Aufgabenfeldes gesehen werden. Trotz der noch nicht ausgeführten und unvollständigen Verbesserungsmöglichkeiten, wie sie in der vorangegangenen Sektion erläutert wurden, kann SyRS als Vorbild für zukünftige Forschungsprojekte dienlich sein. Die Grundkonzepte des modularen Aufbaus, der Trennung des Regelwerks von der Programmierung, der computerlinguistischen Grundlagen und des Einsatzes maschinellen Lernens stellen eine interessante Kombination dar, wie diese Aufgabe handhabbar gemacht werden kann.

Der zukünftige Forschungsbedarf bleibt jedoch beträchtlich: Immer mehr Forschungsgebiete wie die Linguistik, die Mathematik, Informatik u.v.a. beschäftigen sich mit den einzelnen, in diesem Gebiet aufeinanderprallenden Problemstellungen. Vieles davon ist bisher nur theoretisch oder bruchstückhaft vorhanden und muss noch entsprechend umgesetzt werden. Neben der vollständigen morphosyntaktischen Analyse des Wortschatzes sind auch einsetzbare syntaktische, semantische und pragmatische Parser auf Satz- und Textebene, ein ausreichender Umgang mit Unschärfe oder Unvollständigkeiten und letztendlich das Miteinbeziehen von „Weltwissen“ noch Zukunftsmusik [39, 31, 23, 58]. Somit schließt diese Arbeit mit einem Zitat von Galileo Galilei: „*Die Neugier steht immer an erster Stelle eines Problems, das gelöst werden will.*“

ANHANG

# A Einführung in Fuzzy-Sets

Eine Repräsentation der Semantik von Texten anhand von Indextermen führt zu Beschreibungen, die oftmals ungenau und nur teilweise mit der wahren Semantik eines Textes einhergehen. Dadurch ergibt sich auch beim Vergleich von Dokumenten über die vorhandenen Indexterme eine gewisse Ungenauigkeit. Diese teilweise Übereinstimmung kann mithilfe von Fuzzy-Sets, bei der jeder Term ein Fuzzy-Set definiert und jedem Dokument ein Zugehörigkeitsgrad (ein Wert zwischen 0 und 1) zu einem Fuzzy-Set zugewiesen wird, abgebildet werden. Diese Interpretation bildet das Grundkonzept von Fuzzy-Set Modellen für den Einsatz im IR. Im Folgenden werden die zugrundeliegenden Konzepte dieser Theorie vorgestellt.

Die traditionelle Mengentheorie teilt einzelne Elemente $x$ eines Definitionsraumes einer Menge $A$ über die Funktion

$$\mu_A(x) = \begin{cases} 1 & \text{genau dann, wenn } x \in A \\ 0 & \text{genau dann, wenn } x \notin A \end{cases} \tag{A.1}$$

zu.

Das bedeutet, dass ein Element $x$ entweder einer Menge $A$ angehört oder nicht.

Die Fuzzy-Set Theorie [97] hingegen beschäftigt sich mit der Repräsentation von Klassen, deren Elemente nicht eindeutig zugewiesen werden können. Die grundlegende Idee dahinter ist die Verwendung einer Zugehörigkeitsfunktion (Membership-Funktion) in Verbindung mit den Elementen einer Klasse. Diese Funktion weist jedem Element einen Wert zwischen 0 und 1 zu, wobei eine 0 keine Zugehörigkeit

und eine 1 eine absolute Zugehörigkeit des Elements zu dieser Klasse darstellt. Werte zwischen 0 und 1 stellen Elemente der Klasse dar, die nur teilweise dieser Klasse angehören. Durch die Zugehörigkeit zu einem solchen Fuzzy-Set erreicht man einen fließenden anstelle eines abrupten Übergangs (siehe Definition 9).

**Definition 9 – Fuzzy-Set**

Ein Fuzzy-Set $A$ eines Definitionsraumes $U$ wird durch eine Zugehörigkeitsfunktion

$$\mu_A : U \rightarrow [0, 1] \tag{A.2}$$

charakterisiert, die jedem Element $x$ von $U$ einen Wert $\mu_A(x)$ aus dem Intervall [0..1] zuweist. Die Kardinalität eines Fuzzy-Sets wird als Summe aller Zugehörigkeitsgrade der Elemente $x$ dieser Menge durch die Formel

$$|A| = \sum_{x \in U} \mu_A(x) \tag{A.3}$$

angegeben. Ein Fuzzy-Set wird als normalisiert bezeichnet, sobald der Zugehörigkeitsgrad mindestens eines Elements $\mu(x) = 1$ ist.

Die häufigsten Operationen auf Fuzzy-Sets sind das Komplement, die Vereinigung zweier oder mehrerer und der Schnitt zweier oder mehrerer Fuzzy-Sets. Die formalen Definitionen dieser Operationen sind unter Definition 10 zusammengefasst.

Fuzzy-Sets sind hilfreich bei der Repräsentation von vagen und ungenauen Sachverhalten. Deshalb werden und wurden sie schon in vielen Bereichen erfolgreich eingesetzt. Auch im Bereich des IR wird diese Technik immer öfter verwendet.

**Definition 10 – Operationen auf Fuzzy-Sets**

Sei $U$ ein Definitionsraum, $A$ und $B$ zwei Fuzzy-Sets aus $U$, und $\overline{A}$ das Komplement von $A$ relativ zu $U$. Weiters sei $x$ ein Element von $U$. Dann können folgende Operationen festgelegt werden [6, 4]:

$$\text{Komplement: } \mu_{\overline{A}}(x) = 1 - \mu_A(x) \tag{A.4}$$

$$\text{Vereinigung: } \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \tag{A.5}$$

$$\text{Durchschnitt: } \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \tag{A.6}$$

$$\text{Gleichheit (strikt): } A = B \Leftrightarrow \forall x \in U, \mu_A(x) = \mu_B(x) \tag{A.7}$$

$$\text{Gleichheit (graduell): } E(A, B) = degree(A = B) = \frac{|A \cap B|}{|A \cup B|} \tag{A.8}$$

wobei $0 \leq E(A, B) \leq 1$ ist, und $E(A, B) = 0$ bedeutet, dass A und B sich nicht überschneiden.

# Literaturverzeichnis


[1] K. Aas and L. Eikvil. Text categorisation: A survey, 1999.

[2] James Allan. Incremental relevance feedback for information filtering. In *Research and Development in Information Retrieval*, pages 270–278, 1996.

[3] Christine Stöckert Anne Schiller, Simone Teufel. *Vorläufige Guidelines für das Tagging deutscher Textcorpora mit STTS*. Universität Tübigen, 1995.

[4] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, ACM Press, New York, 1999.

[5] Avrim Blum, Adam Kalai, and John Langford. Beating the hold-out: Bounds for k-fold and progressive cross-validation. In *Computational Learing Theory*, pages 203–208, 1999.

[6] Abdelhamid Bouchachia. *Information Retrieval Techniques for Software Retrieval*. PhD thesis, University of Klagenfurt, 2001.

[7] Leigh Star S. Bowker G. C. *Sorting Things Out, Classification and its Consequences*. PhD thesis, Laboratory for Computer Science, Massachusetts Institute of Technology, 1999.

[8] Eric Brill. A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing*, pages 152–155, Trento, IT, 1992.

[9] Eric Brill. Some advances in transformation-based part of speech tagging. In *National Conference on Artificial Intelligence*, pages 722–727, 1994.

[10] Eric Brill. Unsupervised learning of disambiguation rules for part of speech tagging. In David Yarovsky and Kenneth Church, editors, *Proceedings of the Third Workshop on Very Large Corpora*, pages 1–13, Somerset, New Jersey, 1995. Association for Computational Linguistics.

[11] James P. Callan. Document filtering with inference networks. In *Research and Development in Information Retrieval*, pages 262–269, 1996.

[12] James P. Callan, W. Bruce Croft, and Stephen M. Harding. The INQUERY retrieval system. In *Proceedings of DEXA-92, 3rd International Conference on Database and Expert Systems Applications*, pages 78–83, 1992.

[13] Jamie Callan. Learning while filtering documents. In *Proc. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR-98)*, pages 224–231, 1998.

[14] Kaushik Chakrabarti and Sharad Mehrotra. Local dimensionality reduction: A new approach to indexing high dimensional spaces. In *The VLDB Journal*, pages 89–100, 2000.

[15] Soumen Chakrabarti, Byron Dom, Rakesh Agrawal, and Prabhakar Raghavan. Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies. *VLDB Journal: Very Large Data Bases*, 7(3):163–178, 1998.

[16] W. W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. In *ACM Transactions on Information Systems*, volume 17, pages 141–173, 1999.

[17] William W. Cohen. Text categorization and relational learning. In Armand Prieditis and Stuart J. Russell, editors, *Proceedings of ICML-95, 12th International Conference on Machine Learning*, pages 124–132, Lake Tahoe, US, 1995. Morgan Kaufmann Publishers, San Francisco, US.

[18] Rowena Marie Cole. Clustering with genetic algorithms. Master's thesis, Nedlands 6907, Australia, 1998.

[19] D. Cristofor and D. Simovici. An information-theoretical approach to clustering categorical databases using genetic algorithms. In *2nd SIAM ICDM, Workshop on clustering high dimensional data*, 2002.

[20] Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Evangelos Simoudis, Jiawei Han, and Usama Fayyad, editors, *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, Oregon, 1996. AAAI Press.

[21] Christos Faloutsos and Douglas W. Oard. A survey of information retrieval and filtering methods. Technical Report CS-TR-3514, Dept. of Computer Science, Univ. of Maryland, 1995.

[22] P. Fischer. *Computer- und Internet-Lexikon.* SmartBooks Publishing, Kilchberg, 2000.

[23] Günther Fliedl. *Natürlichkeitstheoretische Morphosyntax - Aspekte der Theorie und Implementierung.* Gunter Narr Verlag Tübingen, 1999.

[24] W. B. Frakes and R. Baeza-Yates. *Information Retrieval: Data Structures and Algorithms.* Prentice Hall, Englewood Cliffs, NJ, USA, 1992.

[25] P. Fries. On theme, rheme and discourse goals. In *Advances in Written Text Analysis*, pages 229–249, 1994.

[26] Chawchat Santimetvirul Gareth Jones, Alexander M. Robertson and Peter Willett. Non-hierarchical document clustering using a genetic algorithm. *Information Research*, 1(1), 1995.

[27] José Maria Gómez-Hidalgo and Manuel de Buenaga Rodriguez. Integrating a lexical database and a training collection for text categorization.

[28] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison Wesley, 1989.

[29] Moises Goldszmidt and Mehran Sahami. A probabilistic approach to full-text document clustering. Technical Report ITAD-433-MS-98-044, SRI International, SRI International, 1998.

[30] G. Gonnet and R. Baeza-Yates. *Handbook of Algorithms and Data Structures.* Addison-Wesley, Wokingham, England, 2nd edition edition, 1991.

[31] Ralph Grishman. *Computational linguistics - An Introduction.* Cambridge University Press, 1986.

[32] Marko Grobelnik and Dunja Mladenić. Efficient text categorization, 1998.

[33] M. Halliday. *An Introduction to Functional Grammar.* Edward Arnold, 1985.

[34] John A. Hartigan. *Clustering algorithms.* John Wiley & Sons, New York, London, Sydney, 1975.

[35] Ji He, Ah-Hwee Tan, and Chew-Lim Tan. A comparative study on chinese text categorization methods. In *PRICAI Workshop on Text and Web Mining*, pages 24–35, 2000.

[36] J. Heaps. *Information Retrieval - Computational and Theoretical Aspects.* Academic Press, 1978.

[37] Gallant Stephen I. *Neural Network Learning and Expert Systems.* MIT Press, 1994.

[38] G. Raskinis I. Moulinier and J. Ganascia. Text categorization: a symbolic approach. In *Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval*, 1996.

[39] Lucja M Iwanska and Stuart C. Shapiro. *Natural Language Processing and Knowledge Representation - Language for Knowledge and Knowledge for Language.* AAAI Press / The MIT Press, 2000.

[40] Makoto Iwayama and Takenobu Tokunaga. A probabilistic model for text categorization: Based on a single random variable with multiple values. In *Proceedings of ANLP-94, 4th Conference on Applied Natural Language Processing*, pages 162–167, Stuttgart, DE, 1994. Association for Computational Linguistics, Morristown, US.

[41] Thorsten Joachims. Text categorization with support vector machines: learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[42] Thorsten Joachims. A statistical learning model of text classification with support vector machines. In W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel, editors, *Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval*, pages 128–136, New Orleans, US, 2001. ACM Press, New York, US.

[43] Kasabov Nikola K. *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering.* MIT Press, 1996.

[44] Michael Kluck and Fredric C. Gey. The domain-specific task of CLEF — specific evaluation strategies in cross-language information retrieval. *Lecture Notes in Computer Science*, 2069:48–??, 2001.

[45] T. Kohonen. *Self Organizing Maps.* Springer-Verlag, 1995.

[46] Robert Korfhage. *Information Storage and Retrieval.* John Wiley & Sons, Inc., 1997.

[47] Ravindra Krovi. Genetic algorithms for clustering: a preliminary investigation. In *Proceedings of the TwentyFifth International Conference on System Sciences*, volume 4, pages 540–544. Kauai,HI, IEEE Computer Society Press, Los Alamitos, CA., January 1992.

[48] G. Lakoff. *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind.* The University of Chicago Press, Chicago, 1987.

[49] Savio L. Lam and Dik L. Lee. Feature reduction for neural network based text categorization. In Arbee L. Chen and Frederick H. Lochovsky, editors, *Proceedings of DASFAA-99, 6th IEEE International Conference on Database Advanced Systems for Advanced Application*, pages 195–202, Hsinchu, TW, 1999. IEEE Computer Society Press, Los Alamitos, US.

[50] D. Lewis. *Representation and Learning in Information Retrieval.* PhD thesis, Graduate Schhool of the University of Massachusetts, 1992.

[51] D. D. Lewis. Evaluating Text Categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 312–318. Morgan Kaufmann, 1991.

[52] D. D. Lewis. An evaluation of phrasal and clustered representations on a text cetegorization task. In *Proceedings of the 15th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 37–50, 1992.

[53] D. D. Lewis. Feature Selection and Feature Extraction for Text Categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 212–217, San Mateo, California, 1992. Morgan Kaufmann.

[54] D. D. Lewis. Evaluating and Optimizing Autonomous Text Classification Systems. In E. A. Fox, P. Ingwersen, and R. Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 246–254, Seattle, Washington, 1995. ACM Press.

[55] David D. Lewis and Marc Ringuette. A comparison of two learning algorithms for text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93, Las Vegas, US, 1994.

[56] Chin-Teng Lin and C. S. George Lee. *Neural Fuzzy Systems.* Prentice-Hall, New Jersey, 1996.

[57] W. Mayerthaler. Über die natürliche fokussierbarkeit blonder frauen. In D. Bittner A. Bittner and K.-M. Köpcke, editors, *Angemessene Strukturen: Systemorganisaiton in Phonologie, Morphologie und Syntax*, pages 253–260. Hildesheim-Zürich-New York: Olms, 2000.

[58] Willi Mayerthaler, Günther Fliedl, and Christian Winkler. *Lexikon der Natürlichkeitstheoretischen Syntax und Morphosyntax.* Stauffenburg Verlag Brigitte Narr GmbH, 1998.

[59] Stefano Mizzaro. How many relevances in information retrieval? *Interacting with Computers*, 10(3):303–320, 1998.

[60] I. Moulinier. A framework for comparing text categorization approaches, 1996.

[61] J. Nichols. Functional theories of grammar. *Annual Review of Anthropology*, 13:97–117, 1984.

[62] K. N. Nwogu and T. Bloor. Thematic progression in professional and popular medical texts. In *Functional and Systemic Linguistics: Approaches ans Uses*, pages 369–384, 1991.

[63] M. Nystrand. *The Structure of Written Communication.* Academic Press inc., 1986.

[64] Douglas W. Oard and Gary Marchionini. A conceptual framework for text filtering process. Technical Report CS-TR-3643, University of Maryland, 1996.

[65] Douglas W. Oard, Jianqiang Wang, and Dekang Lin. Trec-8 experiments at maryland: Clir, qa and routing.

[66] D. Osherson and E. Smith. On the adequacy of prototype theory as a theory of concepts. *Cognition*, 9:35–58, 1991.

[67] Luigi Portinale and Lorenza Saitta. Feature selection, 2002.

[68] S. E. Robertson. The probability ranking principle in ir. In *Journal of Dokumentation*, volume 33, pages 294–304, 1977.

[69] Stephen E. Robertson and Ian Soboroff. The TREC 2001 filtering track report. In *Text REtrieval Conference*, 2001.

[70] R. Rojas. *Neural Networks: A Systematic Introduction.* Spring-Verlag, 1996.

[71] Miguel E. Ruiz and Padmini Srinivasan. Hierarchical neural networks for text categorization. In Marti A. Hearst, Fredric Gey, and Richard Tong, editors, *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, pages 281–282, Berkeley, US, 1999. ACM Press, New York, US.

[72] Miguel E. Ruiz and Padmini Srinivasan. Hierarchical text categorization using neural networks. *Information Retrieval*, 5(1):87–118, 2002.

[73] G. Salton. *The SMART Retrieval System - Experiments in Automatic Document Processing.* Prentice Hall Inc., Englewood Cliffs, NJ, 1971.

[74] G. Salton and M. E. Lesk. Computer evaluation of indexing and text processing. *Journal of the ACM*, 15(1):8–36, 1968.

[75] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill Book Co., New York, 1983.

[76] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[77] Robert E. Schapire, Yoram Singer, and Amit Singhal. Boosting and Rocchio applied to text filtering. In W. Bruce Croft, Alistair Moffat, Cornelis J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 215–223, Melbourne, AU, 1998. ACM Press, New York, US.

[78] Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *Konferenz: Textkorpora und Erschließungswerkzeuge.* Universität Stuttgart, 1994.

[79] Helmut Schmid. Improvements in part-of-speech tagging with an application to german. In *Konferenz: EACL SIGDAT.* Universität Stuttgart, 1995.

[80] Hinrich Schutze, David A. Hull, and Jan O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *Research and Development in Information Retrieval*, pages 229–237, 1995.

[81] Fabrizio Sebastiani. Machine learning in automated text categorisation: a survey. Technical Report IEI-B4-31-1999, Instituto di Elaborazione del l'Informatione, Consiglio Nationale delle Ricerche, Pisa, Pisa, IT, 1999.

[82] Fabrizio Sebastiani. A tutorial on automated text categorisation. In Analia Amandi and Ricardo Zunino, editors, *Proceedings of ASAI-99, 1st Argentinian Symposium on Artificial Intelligence*, pages 7–35, Buenos Aires, AR, 1999.

[83] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques, 2000.

[84] A. John Swets. *Signal detection and recognition by human observers.* John Wiley & Sons, New York, 1964.

[85] J. Tague-Sutcliffe. Measuring the informativeness of a retrieval process. In *Proc. of the 15th Anual Int. ACM SIGIR '92*, pages 23–36, 1992.

[86] David Thaler and Chinya V. Ravishankar. Distributed top-down hierarchy construction. In *INFOCOM (2)*, pages 693–701, 1998.

[87] G. S. Wang V. V. Raghavan and P. Bollmann. A critical investigation of recall and precision as measures of retrieval system performance. In *ACM Transactions on Information Systems*, volume 7, 1989.

[88] C. J. Van Rijsbergen. *Information Retrieval, 2nd edition.* Dept. of Computer Science, University of Glasgow, 1979.

[89] A. Vinokourov and M. Girolami. A probabilistic hierarchical clustering method for organising collections of text documents, 2000.

[90] Ellen M. Voorhees. Natural language processing and information retrieval. In *SCIE*, pages 32–48, 1999.

[91] R. Burgin W. M. Shaw Jr. and P. Howell. Performance standards and evaluations in ir test collections: Cluster-based retrieval models. *Information Processing and Management*, 33(1):1–14, 1997.

[92] Erik D. Wiener, Jan O. Pedersen, and Andreas S. Weigend. A neural network approach to topic spotting. In *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 317–332, Las Vegas, US, 1995.

[93] Yiming Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1/2):69–90, 1999.

[94] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In Marti A. Hearst, Fredric Gey, and Richard Tong, editors, *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, pages 42–49, Berkeley, US, 1999. ACM Press, New York, US.

[95] Øystein Grøvlen. Natural language processing in information retrieval, 1995.

[96] Osmar R. Zaïane and Maria-Luiza Antonie. Classifying text documents by associating terms with text categories.

[97] L. A. Zadeh. Fuzzy sets. In H. Prade D. Dubois and R. R. Yager, editors, *Readings in Fuzzy Sets for Intelligent Systems*. Morgan Kaufmann, 1993.

[98] G. Zipf. *Human Behaviour and the Principle of Least Effort*. Addison-Wesley, 1949.