AUTOMATIC INDEXING AND ABSTRACTING OF DOCUMENT TEXTS

Marie-Francine Moens

KLUWER ACADEMIC PUBLISHERS

AUTOMATIC INDEXING AND ABSTRACTING OF DOCUMENT TEXTS

THE KLUWER INTERNATIONAL SERIES ON INFORMATION RETRIEVAL

Series Editor

W. Bruce Croft

University of Massachusetts
Amherst, MA 01003

Also in the Series:

MULTIMEDIA INFORMATION RETRIEVAL: Content-Based Information Retrieval from Large Text and Audio Databases
by Peter Schäuble
ISBN: 0-7923-9899-8

INFORMATION RETRIEVAL SYSTEMS
by Gerald Kowalski
ISBN: 0-7923-9926-9

CROSS-LANGUAGE INFORMATION RETRIEVAL
edited by Gregory Grefenstette
ISBN: 0-7923-8122-X

TEXT RETRIEVAL AND FILTERING: Analytic Models of Performance
by Robert M. Losee
ISBN: 0-7923-8177-7

INFORMATION RETRIEVAL: UNCERTAINTY AND LOGICS: Advanced Models for the Representation and Retrieval of Information
by Fabio Crestani, Mounia Lalmas, and Cornelis Joost van Rijsbergen
ISBN: 0-7923-8302-8

DOCUMENT COMPUTING: Technologies for Managing Electronic Document Collections
by Ross Wilkinson, Timothy Arnold-Moore, Michael Fuller, Ron Sacks-Davis, James Thom, and Justin Zobel
ISBN: 0-7923-8357-5

AUTOMATIC INDEXING AND ABSTRACTING OF DOCUMENT TEXTS

by

Marie-Francine Moens
Katholieke Universiteit Leuven, Belgium

KLUWER ACADEMIC PUBLISHERS
New York / Boston / Dordrecht / London / Moscow

eBook ISBN: 0-306-47017-9
Print ISBN: 0-792-37793-1

Created in the United States of America

Visit Kluwer Online at: http://www.kluweronline.com
and Kluwer's eBookstore at: http://www.ebooks.kluweronline.com

to Peter, Michael, and Laura

This page intentionally left blank.

CONTENTS

PREFACE

Currently, we are confronted with a huge quantity of electronic documents that are written in natural language text. We are good at creating the texts, but not as capable at managing their information content. The documents are stored on computer disks or on CD-ROMS to form large collections. Retrieval systems, search engines, browsing tools, and other information management software are at our disposal for selecting relevant documents or information from the collections. When present-day retrieval and information selection tools operate on the content of document texts or make it accessible, they are not sufficiently powerful to identify documents or information that might be relevant to their users.

Text indexing and abstracting are old techniques for organizing the content of natural language text. These processes create a short description or characterization of the original text, which is called a text representation or representative and has a recognized and accepted format. Indexing commonly extracts from or assigns to the text a set of single words or phrases that function as index terms of the text. Words or phrases of the text are commonly called natural language index terms. When the assigned words or phrases come from a fixed vocabulary, they are called controlled language index terms. The index terms, besides reflecting content, can be used as access points or identifiers of the text in the document collection. Abstracting results in a reduced representation of the content of the text. The abstract usually has the form of a continuous, coherent text or of a profile that structures certain information of the original text.

The idea and the first attempts of automating text indexing and abstracting go back to the end of the 1950s. What at that time was a progressive theory has now become an absolute necessity. The manual task of indexing and abstracting is simply not feasible with the ever expanding collections of textual documents (e.g., on the Internet). Automatic indexing and abstracting, besides being efficient, probably produce a more consistent,

objective and more complete final product. The process of automatic indexing and abstracting starts when the text is already electronically stored and can be regarded as a string of characters (including spaces and punctuation marks). As in the case of manual indexing and abstracting, the automated method entails content analysis of the text, selection and generalization of information, and translation into a final form. Current systems that index and abstract texts generate text representations that are similar to those prepared by humans in terms of content and format (e.g., set of index terms, abstract in the form of a fluent text). This is because retrieval and other text management systems support these representations.

Text representations are used in systems that manage document contents. The majority of them are document retrieval systems. The ultimate goal of indexing and abstracting in text retrieval is an effective retrieval operation, so that more relevant and less irrelevant items are found. It is currently assumed that the major problem in current retrieval systems is capturing the meaning that a document may have for its user. Thus, progress can be made by accurately defining a user's need. We do not deny the importance of an accurate representation of the user's need, but accurately defining information needs will only work well with richer semantic representations of the textual content of documents produced by automatic indexing and abstracting. Current text representations that are automatically generated are only crude reflections of the content of document texts. They are often restricted to some terms that frequently occur in the text, to all words from the beginning of the text, or to sentences that contain frequent terms.

An intuitive solution to generating rich semantic representations of the natural language texts is to analyze them and to interpret their words and phrases based on complete linguistic, domain world, and contextual knowledge. Given the current state of natural language processing, this is not possible, nor is it always desirable. Linguistic knowledge refers to the lexical, syntactic and semantic properties of the texts' language and the typical properties of the discourse. Domain knowledge describes the concepts and subconcepts of the subject domain and their relationships. The contextual knowledge concerns communicative knowledge, which deals with the preferences and needs of those who use information in the texts. A working hypothesis in the domain of information retrieval is that valid text representations can be made without subjecting text to a complete and complex language-dependent processing. This is a valid hypothesis to start with. In the course of this book we will develop and defend a few lesser hypotheses. First, it is stated that knowledge of discourse structures – whether inherent or not to the text type or genre – and of surface linguistic cues that signal them is very useful for automatically indexing and abstracting a text's content. This knowledge also allows us to focus upon certain information in texts that is relevant for specific communication

needs, It is also possible to learn discourse structures from texts with statistical techniques, Finally, domain knowledge is important to identify topical concepts in texts. Knowledge of concepts and their variant textual patterns can be learned from example texts.

The book has ambitious objectives: to study automatic indexing and abstracting in all its facets and to describe the latest novel techniques in automatic indexing and abstracting. In addition, it confronts the many problems that automatic indexing and abstracting of text pose. Although, the book focuses upon indexing and abstracting of written text, many findings are also important for spoken textual documents, which are increasingly used for communication and storage of information.

This book is organized as follows:

The first part, "*The Indexing and Abstracting Environment*", places the problem in a broad context and defines important concepts of the book. The first chapter, "*The Need for Indexing and Abstracting Texts*", justifies the urgency for better methods for automatic indexing and abstracting of text content. From a broad viewpoint, some pertinent problems in information retrieval and text management in general are discussed. The current solutions to these problems are outlined. In the course of this chapter, the real need for better automatic indexing and abstracting techniques becomes clear. The second chapter of this part, "*The Attributes of Text*", elaborates on the features of text. It gives an overview of the different components and structures that make up a text. The last chapter of this part, "*Text Representations and their Use*", discusses the properties and use of different text representations for document and information retrieval.

The second part of the book, "*Methods of Automatic Indexing and Abstracting*", gives an overview of existing techniques of automatic indexing and abstracting. Currently, such a detailed overview is lacking in the literature. The different chapters deal with the major forms of text representations: *"Automatic Indexing: The Selection of Natural Language Index Terms"*, *"Automatic Indexing: The Assignment of Controlled Language Index Terms"*, and *"Automatic Abstracting: The Creation of Text Summaries"*. The content of this part provides the context for the applications discussed in the third part and justifies the choice of certain techniques in the applications.

The third part of the book considers "*Applications*". Four important problems are described for two collections of texts, written in Dutch. The problems mainly regard indexing with controlled language index terms, text classification, and abstracting. One corpus contains the texts of legal cases, while the other is composed of magazine articles. Solutions are proposed and tested with the help of software for indexing and abstracting, which the author designed and implemented. The applications elaborate on novel techniques and improve existing ones for automatic indexing and

abstracting. The first chapter "*Text Structuring and Categorization when Summarizing Legal Cases ",* deals with a successful initial categorization and structuring of the criminal cases. A text grammar is employed to represent knowledge of case structures, of concepts typical for the criminal law domain, and of the information focus. In the next chapter, "*Clustering of Paragraphs when Summarizing Legal Cases"*, a number of lengthy passages of the legal cases are summarized by extracting representative paragraphs and key terms. The techniques for identifying the representative textual units rely upon the distribution of lexical items in the legal texts and demonstrate the usefulness of clustering based on the selection of representative objects. In the third chapter entitled "*The Creation of Highlight Abstracts of Magazine Articles"*, the portability of the text grammar approach for text abstracting is demonstrated, in the process of creating highlight abstracts of magazine articles. Here, the typical discourse patterns of news stories are taken advantage of. In the last chapter of this part, "*The Assignment of Subject Descriptors to Magazine Articles "*, the technique learns the typical text patterns of the broad subject classes of the articles from a limited set of example texts and applies this knowledge for assigning subject descriptors to new, previously unseen articles.

The book concludes with a summary, an overview of the contributions of the research, and directions for future research.

The book is interdisciplinary. Its subject, "*Automatic Indexing and Abstracting of Document Texts*", is an essential element of information retrieval research. Information retrieval is a discipline that has its foundations in information science, computer science, and statistics. The research especially studies text and its automatic analysis. This is the research domain of computational linguistics, a subdiscipline of computer science. Because of the nature of the two text corpora used in the research, legal texts and magazine articles, the research encounters the disciplines of law and communication science. The field of cognitive science is touched upon when the cognitive process of indexing and abstracting yields models for the automatic processes.

ACKNOWLEDGEMENTS

This publication is a slightly shortened version of my doctoral dissertation defended on June 28, 1999 in the Faculty of Sciences at the Katholieke Universiteit Leuven, Belgium. Though it is impossible to acknowledge the contributions of those who have helped me, I would like to mention those whose assistance was direct and vital to the completion of this work.

The far origins of this book lie in my work on ancient Egyptian language guided by Professor J. Quaegebeur (Katholieke Universiteit Leuven, Belgium) and Professor J. Callender (University of California Los Angeles, California, U.S.A.) who have awaken in me a profound interest in the analysis of language and texts.

I am very grateful to Professor J. Dumortier, my supervisor, who gave me the magnificent chance to explore the subject of this book. He gave me the opportunity to work at the Interdisciplinary Centre for Law and Information Technology (ICRI) (Katholieke Universiteit Leuven, Belgium), which is a very stimulating environment for creative research. It was under his supervision that the research contained in this volume started some five years ago.

I must express my gratitude to the advisors of my doctoral dissertation at the Katholieke Universiteit Leuven, Belgium: Professor H. Olivié, Professor L. Verstraelen, and Professor J. Dumortier. Their continuous encouragement have greatly facilitated its preparation. I thank Professor H. Olivié for his helpful advice.

I also thank the members of the examination jury, Professor D. De Schreye (Katholieke Universiteit Leuven, Belgium), Professor J. Leysen (Koninklijke Militaire School, Belgium), and Professor J. Hobbs (Stanford Research Institute, California, U.S.A.), who by their remarks and suggestions allowed me to achieve the final goals of this publication.

It is with deep respect that I thank Professor A. Oosterlinck, Rector of the Katholieke Universiteit Leuven, Belgium, and Professor J. Herbots, Dean of

the Faculty of Law for having given me the opportunity to work at the Katholieke Universiteit Leuven, Belgium. I must also thank Professor J. Berlamont, Dean of the Faculty of Applied Sciences, who has given me the opportunity of pursuing a doctoral training in Computer Science at the Katholieke Universiteit Leuven, Belgium, and Professor L. Vanquickenbome, Dean of the Faculty of Sciences, who allowed me to defend my doctoral degree. I thank Professor S. Vandewalle for taking care of my dossier regarding the doctoral training.

I am most indebted to my colleague Drs. C. Uyttendaele who provided invaluable help in one of the projects described in the book and who translated most of the legal texts from Dutch to English. I am also grateful to Mrs. T. Bouwen for the verification of some of the results contained in this publication. I thank Dr. W. Wetterstrom (Harvard University, Massachusetts, U.S.A.) who helped me correcting my English in the preface and summary. I wish also to thank Professor J. Zeleznikow (La Trobe University, Australia) for his helpful comments. I thank the anonymous reviewers of my research papers that are integrated in this book.

Additionally, I am grateful to Dr. C. Belmans and Ir. J. Huens (Katholieke Universiteit Leuven, Belgium) and Mr. L. Misseeuw and Mr. P. Huyghe (Roularta Media Group) for their technical assistance in making the text corpora available. I am grateful to Mrs. N. Verbiest for the administrative support. I wish to thank my family and colleagues for their continuous encouragement.

Finally, I would like to express my gratitude to the organizations that provided me with grant support during my studies and research: the Belgian American Educational Foundation (BAEF), the Ministerie van Onderwijs Bestuur van het Hoger Onderwijs en het Wetenschappelijk Onderzoek, the Onderzoeksfonds K.U.Leuven, the Nationaal Fonds voor Wetenschappelijk Onderzoek (NFWO), the Vlaams Instituut voor de bevordering van het Wetenschappelijk-Technologisch onderzoek in de industrie (IWT), the Vlaamse Leergangen Leuven, and the Vlaamse Wetenschappelijke Stichting.

PART I

THE INDEXING AND ABSTRACTING ENVIRONMENT

This page intentionally left blank.

Chapter 1

THE NEED FOR INDEXING AND ABSTRACTING TEXTS

1. INTRODUCTION

People communicate by conversing. Since early times, mankind employs recorded forms of communication. One of them, written text is generally considered as marking the historical era of mankind. People learned to code the audible utterances into sequences of graphical symbols, and to decode the writing again in terms of spoken language. Even, though text written in natural language is only a crude form of representing what goes on in the mind of the writer, it serves an important role in communication. Recent developments in electronic technology have introduced many new physical forms of communication, but have not stopped the production of documents in the form of written texts. Technology not only accounts for their easy creation, but also for their unrestrained reproduction and dissemination. However, the crucial concern is an effective dissemination of electronic documents. When people are confronted with large electronic document bases, they want to find the documents and information relevant to their needs.

This chapter explains some important concepts and problems of document and information[1] selection in general and of text retrieval in specific. It gradually shapes the affirmation that there is a definite need for automatic indexing and abstracting with advanced text analysis methods without invoking complex and complete natural language processing of the texts. Tools for indexing and abstracting the content of texts are necessary

components of future information retrieval and selection systems. They will complement the tools for analysis of image data and speech recognition in managing the content of documents.

2. ELECTRONIC DOCUMENTS

The concept "*document*" is used as a noun as well as a verb. The Latin word "documentum" means "official paper used as a piece of evidence or proof, in some cases to be taken as an example". In its narrow sense, the noun document still has this association (e.g., a contract). In the course of history, the concept document is used in a broader sense being: "any printed representation containing text and/or non-textual components such as photo's, signatures, charts, tables, etc., which is produced with the intention to share knowledge" (Vervenne, Hamerlinck, & Vandamme, 1995). The verb "to document" means to illustrate or to show evidence. In its broader sense, the verb refers to all actions dealing with the editing, the printing, and the distribution of documents. From this point of view, a document is an important interpersonal and social means of communication between its creator and its user (Schamber, 1996). The creator uses the document content to describe, organize, and synthesize his ideas. He purposefully creates the document in a manner that its users can understand its contents in the most optimal way. For effective communication, the document must provide information that contributes to a user's work or interest.

In our current society documents based on paper and print are gradually replaced by *electronic documents.* Electronic documents are stored on electronic media such as CD-ROMS or distributed hardware disks accessible through networks (e.g., Internet). Electronic documents have some important characteristics (for more details see Schamber, 1996):

1. They are easily created, manipulated and unrestrainedly replicated by authoring systems. They are also easily transportable and efficiently stored. As a result, we are confronted with massive volumes of electronic documents.
2. They can be remarkable elusive, transient and constantly evolving. On the other hand, they are available simultaneously for many people.
3. They create new communicative structures and open vistas for new regularized codification and notation systems (e.g., mark-up languages) that allow representing new types of content (e.g., video and audio data in multimedia documents).

3. COMMUNICATION THROUGH NATURAL LANGUAGE TEXT

Many current documents contain natural language text. Natural language text is highly valued as a means of communication. Defining the concepts of communication and text clarifies why they are tightly related.

Communication has been studied extensively and different models of communication have been proposed. Communication involves a sender and a receiver. In the case of communication by means of a document, we speak of a creator and a user. In the *code model* (Shannon & Weaver, 1949), which goes back to Aristotl (Sperber & Wilson, 1995, p. 2), communication is achieved by encoding a message, which cannot travel, into a signal, which can travel, and by decoding the signal at the receiving end. Such a view implies the *mutual-knowledge hypothesis.* This hypothesis states that if the receiver is to be sure of recovering the correct interpretation, the one intended by the sender, every item of contextual information used in interpreting the message must be known mutually by sender and receiver. Sperber and Wilson (1995) regard verbal communication or communication in natural language as involving two types of communication processes: one based on coding and decoding, the other on ostension and inference. Acoustic or graphic signals are used to communicate semantic representations. The semantic representations recovered by decoding are useful only as a source of hypotheses and evidence for the second communication process the inferential one. According to the *ostensive-inferential model*, communication is achieved by the communicator providing evidence of his or her intentions and by the audience inferring his intentions from the evidence. The communicator makes his or her communicative intentions or goals ostensive, while signaling a public interpretation of his or her thoughts. Ostension helps to focus the attention of the audience on the relevant information. The audience applies inference rules to the recovered semantic representations of the thoughts of the communicator to form a mental interpretation of them. This interpretation goes as far as inferring a meaning that was not meant by the communicator. Mutual knowledge is certainly involved in verbal communication, but the communication aims at enlarging and modifying the mutual cognitive environments of communicator and audience, and does not direct at duplicating thoughts.

Text is defined by Petöfí and Garcia Berrio (1978, cited by Pinto Molina, 1995) as “a group of linked linguistic units in a total conglomerate of communicative intention”. De Beaugrande and Dressler (1981, p. 3 ff.) define text as a *communicative occurrence* that meets seven standards of

textuality. The first standard, *cohesion*, concerns the ways in which components of the surface text, i.e., the actual words (language expressions) we hear or see, are mutually connected within a sequence. The surface components depend upon each other according to grammatical forms and conventions. Cohesion bears on the connectivity of the surface expressions. The second standard, *coherence*, concerns the ways in which the components of the textual world, i.e., the configuration of concepts and relations that underlie the surface text are mutually accessible and relevant. Coherence regards the global organization and connectivity of the underlying content. Cohesion and coherence are text-centered notions (see chapter 2). The remaining standards are user-centered notions, which are brought to bear on the activity of textual communication at large, both by creators and users. *Intentionality* concerns the creator's attitude that a set of occurrences should constitute a cohesive and coherence instrumental in fulfilling the creator's intentions, e.g., to distribute knowledge or to attain a specified goal. *Acceptability* pertains to the text user's attitude that the set of occurrences should constitute a cohesive and coherent text having some use or relevance for the user, e.g., to acquire knowledge or provide co-operation in a plan. *Informativeness* concerns the extent to which the occurrences of the presented text are expected vs. unexpected or known vs. unknown. *Situationality* relates to the factors that make a text relevant to a situation of occurrence. The last standard, *intertextuality*, concerns the factors that make the utilization of one text dependent upon knowledge of one or more previously encountered texts. Intertextuality is responsible for the evolution of text types as classes of texts with typical patterns of characteristics.

So far, it is clear that text makes a whole range of communicative activities possible. Text is closely related to *natural language*. Its content is mainly manifested by natural language expressions. Natural language is the most elaborate symbolic system that human beings control and is an essential tool in many cognitive processes including communication, and processing and memorizing of information (Sperber & Wilson, 1995, p. 173). The representation power of natural language is unrivaled. Natural language provides an economical, effective and expressive tool for communication of content (Sparck Jones, 1991). The individual words in a text and their ordering manifest the content of that text. It is unlikely that natural language will be given up in favor of an artificial language for expressing a text's content (Coulmas, 1989, p. 27). According to Coulmas, it might be possible for a group of people to develop a graphical code which is independent of their natural language and which reaches the same complexity and expressive power as their language. However, it would be highly unlikely the coding would be used in human communication.

Text manifests itself in a spoken (speech) as well as in a written form (Figge, 1979). In this book, we concentrate on written text.

4. UNDERSTANDING OF NATURAL LANGUAGE TEXT: THE COGNITIVE PROCESS

Text can be considered as a complex cognitive and social phenomenon. Psychologists have studied the cognitive process of text comprehension or understanding. Pioneers in this research are *Kintsch and van Dijk* (1978; see also van Dijk & Kintsch, 1983). Kintsch and van Dijk assume that, when reading a text, its surface features (words and their ordering in the text) are interpreted as a set of propositions. A proposition is a common form for representing the content of a sentence. Various semantic relations among the propositions order this set. Some of these relations are explicitly expressed in the surface structure of the discourse; others are inferred during the process of interpretation with the various kinds of context-specific or general knowledge. From this set of ordered propositions, the general subject or topic is inferred. Conventional production schemata of texts help in specifying the kind of information that is important in a particular comprehension task. According to Kintsch and van Dijk (1978) text has a number of structures that allow us to comprehend the text and identify the content of the text. In chapter 2 we will elaborate on these text structures.

Since the publication of the notorious Kintsch and van Dijk paper (1978), multiple cognitive studies have affirmed that the cognitive process of understanding engages many knowledge sources and sustains multiple inferences. These studies also emphasize that understanding a text also involves attaching a personal meaning or interpretation to it, which is not exclusively embedded in the text itself. The model of *Graesser and Clark* (1985, p. 14 ff.) relates four knowledge sources to text understanding. The first source is the explicit linguistic material, including words, syntactic constructions, and linguistic signaling devices that are explicitly manifested in the text. It also includes the linguistic knowledge that the comprehender has about these levels of language analysis. The second source consists of world knowledge structures that are stored in the comprehender's long-term memory. These knowledge structures include both generic knowledge structures and specific knowledge structures. Comprehension suffers when the comprehender's knowledge of the words and topics of the text is inadequate. The third source consists of the goals of the comprehender who reads the text. The meaning of a text varies when a text is accessed for different purposes. The fourth source consists of the pragmatic context of the

communication. This includes the social relationship between the reader and the writer, the shared knowledge between the participants of the communicative event, and socially shared attitudes and ideologies. Many inferences are generated during text comprehension, if the comprehender's knowledge base is very rich, and reasoning strategies vary from knowledge domain to knowledge domain (Schank, 1982; Graesser & Clark, 1985, p. 15 ff.). Inferences depend upon knowledge to be found in the text (e.g., the meaning of other, mostly previous sentences), the user's general knowledge system, and upon the purpose of reading a text (Black, 1981; Shiro, 1994). Text understanding involves a huge amount of contextual information. Psychological efforts have not yet converged on a clear picture of what inferences are generated, and how many inferences are generated. More studies are needed to describe which reasoning strategies are employed in different knowledge domains.

Current research emphasizes the need for models of text understanding that involve the *subjective model of the reader* (van Dijk, 1995). Because text understanding is personal, ad hoc and unique, and would define one specific interpretation of a specific text at a specific moment, a model of text understanding would feature personal associations, inferences, and context.

5. UNDERSTANDING OF NATURAL LANGUAGE TEXT: THE AUTOMATED PROCESS

The complexity of the cognitive process of understanding natural language text makes the automation of this process a very challenging task. Automatic understanding of texts belongs to the research field of natural language processing. *Natural language processing* aiming at a fully-understood interpretation of texts deals with processing the linguistic coding (vocabulary, syntax, and semantics of the language and discourse properties), domain world knowledge, shared knowledge between the creator and user of the text, and the complete context of the understanding at a specific moment in time, including the ideology, norms, background of the user, and the purposes of using the text. The processing would not only reveal the content of the text, it would also clarify the meaning that the text has for its user.

Such a full understanding of texts including its interpretation is far from realized by automatic means. The problems of automatic text understanding concern both the modeling of the knowledge and the inference mechanism involved, and the computational complexity of the operations. Besides the enormous task of *acquiring the knowledge and inferences* needed – many of

the relevant structures and strategies involved are still unknown (van Dijk, 1995) – there is the ambitious task of designing workable models. Especially the knowledge about goals, believes, values, and emotional states of a user of the information in the text and the whole pragmatic context of the communication are very hard to model. Moreover, the model must be able to adjust to changes in the personal situation of the text user. Besides the problem of an exhaustive and correct modeling of the knowledge and inference processes involved, researchers worry about the *computational complexity* and the potential problems when different knowledge structures interact (Jacobs & Rau, 1993).

So, the complex expressive and communicative power of natural language texts makes them at present not yet completely understandable by machine. Research in automatic language understanding has focused on linguistically restricted input and on task driven interpretation of texts.

The term "*sublanguage"* is used when texts deal with a restricted subject domain and are processed for specific purpose results. The term is still more appropriate when a community of text creators and users sharing specialized knowledge uses the sublanguage. Such a sublanguage is more restricted in its linguistic properties (vocabulary, syntax, semantics, and discourse organization) (Kittredge & Lehrberger, 1982; Grishman & Kittredge, 1986). Typical sublanguage texts may be weather reports and medical discharge summaries of patients. However, linguistic expressions from the standard language or from neighboring domains possibly enter the sublanguage without going through a process of setting up conventions. The desire to automatically manipulate such a sublanguage inevitably leads to the prescription of additional constraints upon language use beyond those inherent in the sublanguage. In a far reaching form such a controlled language can develop towards a complete "artificial language", which misses the expressive and communicative power of a natural language.

A second approach related to the foregoing regards *task driven interpretation* of texts (Jacobs & Rau, 1993). When a text is employed with clear goals shared among its users, its processing focuses upon identification of specific information in it, while neglecting its complete understanding. The information in focus typically has a meaning for a class of users. Such an approach necessarily reduces the complexity of the text understanding process.

Indexing and abstracting are old techniques for organizing the content of natural language text. These processes create a short description or characterization of the original text, which is called a text *representation. Indexing* commonly extracts from or assigns to the text a set of single words or phrases that function as index terms of the text. *Abstracting* commonly

creates a short coherent text or a profile that structures certain information of the original text. Simple automatic methods aim at identifying topic terms based on occurrence frequencies of individual words in text and reference corpora. In case of abstracting, sentences are extracted that contain important topic terms. This shallow form of text understanding is much in use for characterizing the content of a document text in current information retrieval and selecting tools (see below).

6. IMPORTANT CONCEPTS IN INFORMATION RETRIEVAL AND SELECTION

Document texts are an important means of communication. Current text processing tools allow for their unconstrained creation and reproduction. As a result, large and constantly evolving collections of texts are at our disposal. Information retrieval and selection tools help in finding documents or information that is relevant for a specific need. These tools mainly regard information retrieval systems, question-answering systems, and browsing systems (Figure 1). A typical *information system* consists of a database of documents, a search engine that identifies the documents or information relevant for the information need, and an interface which allows expressing an information need (*query* or *question*), consulting the search results, or browsing the collection.

Document or information retrieval is concerned with selecting documents that the user wants to read to learn something about it. Despite the emerging interest in sound and image retrieval, the term *"text retrieval"*, referring to the process of retrieving textual documents, is often seen as a synonym for document retrieval (Lewis & Sparck Jones, 1996). The basic *process of information retrieval* can be described as follows: representing a user's information problem or need, representing the content of documents, and comparing these representations to decide which documents best correspond to the information need and should be retrieved. As we will further explain in chapter 3, correspondence is found through matching or inference. Often, documents and natural language queries are represented in an abstract form facilitating the matching between document and information need. *Document filtering* and *routing systems* operate in the same fashion, but the information need is generally more stable and long-termed.

Question-answering systems, which we also call *text extraction* systems, involve the retrieval of information and knowledge from the texts of documents (Lewis & Sparck Jones, 1996). A text extraction system usually analyzes volumes of unstructured text, selects certain features from the text

and potentially stores such features in a structured form (Jacobs, 1992, p. 2). So, a collection of structured document surrogates or representations can replace the document collection. The extracted information and knowledge form the answers to specific questions posed to the document texts. As we will further explain in chapter 3, correspondence is found through matching or inference.

In *browsing or navigation systems* there is no information need. Browsing systems are usually part of hypertext and hypermedia systems. *Hypertext and hypermedia systems* (Conklin, 1987; Nielsen, 1995) store and manage document collections, which respectively contain textual items and many other different digitized forms of media. Usually, a document is split into parts or fragments. All fragments are stored and managed in a network of nodes, where each node of the network contains a fragment and related nodes are connected through connections called information links. Documents and their parts are interconnected in this way. Each sequence of

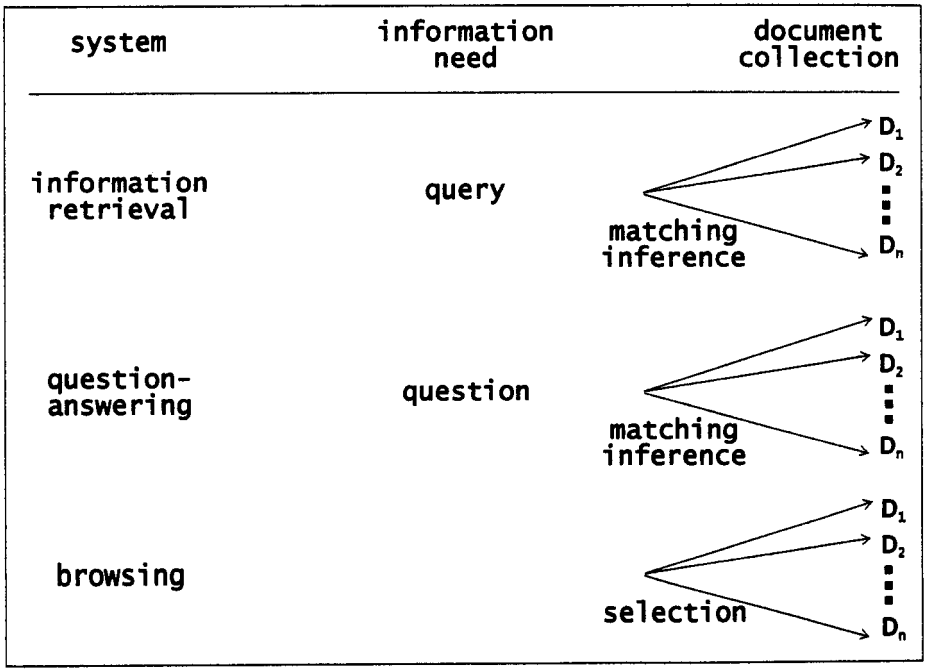


Figure 1. Document and information selection tools.

connections forms a different path for consulting (navigating in) the document or document collection. In this way, a collection can be explored in a non-sequential way (e.g., jumping from one text to another). A user selects documents by browsing their full-texts or by browsing their abstracts (Croft, 1993).

When current information retrieval and selection tools operate on the content of document texts, they are insufficiently effective to identify documents or information that is relevant for their users. In the following, we will explain the concepts of aboutness, relevance and information need. These concepts are fundamentally related and need an explanation in order to explicate fully the information retrieval problem. We use here the term "information retrieval" as a general term for information and document selection.

6.1 Aboutness and Meaning

The *aboutness* or *topicality* of a text regards the subjects or topics discussed in the text (Schank, 1982; Beghtol, 1986). A text has a relatively permanent aboutness and the aboutness is usually agreed upon among the different actors in a communication process (creator(s) and user(s) of the text). The aboutness of a text is not always explicitly stated by the surface features of the text, it possibly involves knowledge that is shared by the creator(s) and user(s) of the text. The above Kintsch and van Dijk model of text understanding (1978) especially aims at understanding the aboutness of a text.

As it is already explained, text comprehension is influenced by many cognitive factors, among which are interest, task, purpose, knowledge, norms, opinions, or attitudes. These factors determine the *meaning* that a text has for its user. Another term sometimes used to indicate the meaning of a text is *interpretation.* Text interpretation consists in general, of reading the text not in a "neutral" fashion with the purpose of single comprehension, but refers to reading the text while considering the whole background situation of the reader or user (Bánréti, 1981). A large amount of textual meaning is constructed by inferences that are made as a result of the interaction between the reader and the text (Shiro, 1994). Meaning may, but not necessarily, refer to *informativeness* (Boyce, 1982). Informativeness is the quality of adding new information to the information that a text user already has. Informativeness and meaning change over time.

A text has an intrinsic subject, an aboutness, but has a variable number of meanings in accordance with the particular use that the person can make of the aboutness at a given time.

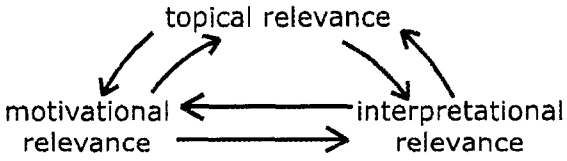


Figure 2. Relationships between topical relevance, motivational relevance, and interpretational relevance (cf. Saracevic, 1975).

Recognition of the relatively permanent quality of aboutness in documents is one of the assumptions upon which bibliographic classification systems have traditionally been based. Aboutness is what a human classifier determines during aboutness analysis of a document, and meaning is the reason a user wants to retrieve it. There is of course, a strong relationship between a document's aboutness and its potential meanings for individuals (Beghtol, 1986). The purpose of retrieval systems is to retrieve documents whose aboutness suggest that a user may find in them meaning(s) expedient to a certain need of the moment. It is interesting to cite the distinction made by Maron (1977) between objective aboutness (what we call here aboutness), subjective aboutness (meaning), and retrieval aboutness, the last referring to the meaning of a text to a class of individuals.

6.2 Relevance

Relevance is a measure of the effectiveness of a contact between a sender and a receiver in a communication process. Relevance likewise deals with the effectiveness of the communication in information retrieval and plays a crucial role in the evaluation of the information retrieved. Relevance in information retrieval is multifaceted. Criteria of relevance in general refer to the information content of documents, to the user's interpretation of the information content, and to the user's motivation when accessing the documents.

Relevance is the relationship of a document to a user's need that it helps to resolve. Prominent among the facets of relevance is *topicality* or aboutness (Schutz, 1970, p. 26 ff.; Saracevic, 1975). Topicality regards the information content of a document and concerns the theme or subject considered in the document. The main theme of a text is an unlimited field for further thematizations. This subthematization involves the enlarging or

deepening of the prevailing theme or the shift from one subtopic to another when there is no hierarchical relation between them. A document may contain a number of subtopics, which in one way or another are relevant to the user. Topicality is not the sole relevance factor related to content, there are other factors that somehow are related to the content such as the depth and the scope of the information, the accuracy of the information, and the situational factors of the reputation of the source and the recency of the information (Barry, 1994).

Besides topical relevance, Schutz (1970, p. 35 ff.) and Saracevic (1975) refer to interpretational relevance and to motivational relevance (Figure 2). *Interpretational relevance* involves the user's interpretation of the document based on his or her own prior experiences, perceptions, or beliefs. Interpretational relevance includes novelty and understandability of the information to the user. *Motivational relevance* includes the purpose of the search and the intended use of the information.

Topical, interpretational and motivational relevance are *interrelated.* Interpretational and motivational relevance involve the meaning of a document to the user and interact dynamically during the relevance assessing process. Topicality refers to the aboutness of a document and plays a significant role in determining the meaning of a document (Boyce, 1982; Beghtol, 1986).

Considering these relevance criteria, it is yet impossible under almost all sets of circumstances to identify precisely and completely the subset of information or documents relevant to a given user in the context of a specific need. First, relevance is a subjective concept depending upon the individual user (Schutz, 1970, p. 35 ff.; Saracevic, 1975; Schamber, Eisenberg, & Nilan, 1990; Barry 1994). Yet, it is very difficult to control an individual's mind at a given moment (cf. Sperber & Wilson, 1995, p. 118 ff.). Second, relevance changes over time depending upon the user's knowledge level and beliefs (Schamber et al., 1990; Barry, 1994). In this strong sense, *assessing of relevance* implicates *measuring the meaning of the document to an individual user at a given time.* So, this strong sense of relevance cannot yet be incorporated wholesale into information retrieval system design and evaluation. Moreover, there is the far-reaching *knowledge-synthesis problem* (Green, 1995). Concomitant with the current information explosion is an increasing trend toward knowledge specialization and fragmentation. So, it is possible that two documents, each of them separately are not relevant for a user's need, but, from their combined use, a solution to the user's need could be inferred.

Consequently, we use several *weaker notions of relevance,* based on the set of operational assumptions underlying a theory of retrieval. The goal of

the ideal document system is, minimally, to identify the document(s) that potentially help a user with respect to his or her need(s) (Green, 1995). In this view relevance is the property of a document's *being potentially helpful to a user in the resolution of a need.* Topical relevance is a necessary, but not sufficient condition for relevance (Froelich, 1994). Topical relevance usually acts as a first filter in selecting documents (Boyce, 1982). It is the easiest relevance factor to deal with in text-based systems and it is the major factor when ranking documents according to their relevance to the query in current information retrieval systems.

Relevance is difficult to compute in exact numbers. Relevance is assessed by people in abstract relative terms (e.g., "weakly relevant", "very relevant", "totally irrelevant"), but not in terms of quantitative absolute judgments and especially not in binary yes-no decisions (Saracevic, 1995). However, performance of information retrieval systems is usually measured in terms of effectiveness metrics, i.e., recall and precision, which rely on a binary relevance judgment of documents. *Recall* measures the proportion of relevant documents retrieved and *precision* the proportion of retrieved documents that are relevant (Salton, 1989, p. 248).

6.3 The Information Need

In current information retrieval systems, an *information need* is usually expressed by key terms or by a Boolean combination of key terms. Or, the query is expressed as a natural language utterance, which is automatically indexed to provide the necessary key terms for document matching. This is, however, a *poor representation* of the real information need. An information need situation encompasses all factors that the user brings to the situation: previous knowledge, awareness of information that is available, affective and emotional factors, the expected use of the information, and other personal and situational factors. Even, when the need is more or less adequately expressed in natural language, its representation is usually reduced to some key terms, which insufficiently represent the real need.

Moreover, the information need situation is *dynamic and constantly changing* (Barry, 1994). Sometimes, the user of a document database does *not* have *a well-defined need.* He or she wishes to skim through the database. Or more strongly, a document only becomes of great importance after completely reading it (Allen, 1990).

It is *very hard to correctly and adequately conceptualize and represent the real information need* of a person at a given time. Nevertheless, given the large number of documents in current document bases, information selection

is necessary. The user does not want to read the fill-text of every document in the collection to satisfy his or her information need.

6.4 The Information (Retrieval) Problem

The core of information retrieval is the problem of evaluating the value of the content of a given document to a given information need. Clearly, the straightforward approach of first understanding the content of the document, and then matching the content with a faithful model of the user's interest, is fraught with daunting problems. The most important problem is the natural language understanding of document texts and user's preferences.

The *process of information retrieval* consists of *several probabilistic operations* (cf. Blair, 1990, p. 319). First, the representation of the information need is often an approximation of the real need of the user or users' group. Second, the natural language understanding of the document text is poor, and often yields an incomplete or incorrect characterization of the text and of its aboutness. Finally, the matching between query and document is a probabilistic operation. Documents are usually ranked according to their probability of relevance to the query. The matching is commonly restricted to a term matching between query and document, whereby the probability of relevance is proportional to the number of matched terms (cf. Green & Bean, 1995). As a result, it is probable that the whole information retrieval operation does not yield all the documents relevant to the query and/or does supply documents that are not or only marginally relevant to the query.

The above problem regards a classical information retrieval system. However, the information problem is also present in browsing systems and in question-answering systems. In browsing systems, the user does not make his information need explicit. However, the systems exhibit a definite need for adequate condensed descriptors of their documents' content (e.g., in the form of topic maps, abstracts, and suggested links), which must guide the user in his or her selection of documents. Then, the information problem regards an inadequate selection of documents due to an incorrect or incomplete characterization of the texts and their aboutness. In question-answering systems, the information need is clearly stated (question for specific information). Here again, the information problem regards the often-faulty characterization of the document content.

7. GENERAL SOLUTIONS TO THE INFORMATION RETRIEVAL PROBLEM

In this section we explain a number of major strategies that have been implemented and still are being developed to relieve the above information retrieval problem.

7.1 Full-Text Search and Retrieval

The basic concept of *full-text search* and retrieval is storing the full-text of all documents in the collection so that every word in the text is searchable and can function as keys for retrieval. Then, when a person wants information from the stored collection, the computer is instructed to search for all documents containing certain user's specified words or word combinations. This approach contrasts with searching collections that have fixed descriptors attached to the document texts.

The original idea (Swanson, 1960) was positively tested by Salton (1970) and since then implementation of full-text retrieval gained more and more success. Today, the full-text segment is still a growing section of the commercial computerized database market (Sievert, 1996).

Full-text search is attractive for many reasons and has some definite *advantages.*

1. Full-text search is attractive from the commercial point of view (Blair & Maron, 1985). Digital technology provides cheap storage for full-text and supplies fast computational technology making searching of full-text efficient. It is also very convenient to search different text types in large document collections just by searching individual words. Additionally, as it employs a simple form of automatic indexing, it avoids the need for human indexers, whose employment is increasingly costly and whose work often appears inconsistent and less fully effective.
2. Full-text search is a first attempt to transfer indexing from a primarily a priori process, to a process determined by specific information needs and other situational factors (Tenopir, 1985; Salton, 1986). Fixed text descriptors severely hamper the accessibility of the texts. Sometimes documents are not retrievable relying on assigned descriptors, because their information value to the users is peripheral to their main focus. Indexing of concepts and terms in a full-text search is situation dependent and would be performed according the requirements of each incoming request.

3. Inexperienced users found that searching with natural language terms in the full-text was easier than searching with fixed text descriptors (Tenopir, 1985).

Still, full-text search is not a magical formula and it suffers from *shortcomings.*

1. While recall is generally enhanced compared to the use of fixed text descriptors (Tenopir, 1985; McKinin, Sievert, Johnson, & Mitchell, 1991), when searching large document collections, precision may suffer intolerably and users might be swamped with irrelevant material (Blair & Maron, 1985; Blair & Maron, 1990). The occurrence of a word or word combination is no guarantee for relevance. As databases grow, this "too many hits" problem will only be exacerbated. This is currently the case with full-text searches on the Internet.
2. Also recall may suffer. A survey by Croft, Krovetz, and Turtle (1990) indicates that users often query documents in terms that they are familiar with, and these terms are frequently not the terms used in the document itself. This shortcoming is still more prominent, when combinations of search terms are used that need to occur together in documents (Blair & Maron, 1985). If the occurrences of these terms in a relevant document are independent events, the probability of finding documents that contain the exact term combination decreases as the number of search terms in the combination increases.

In the past years, research on full-text retrieval has increased dramatically because of the yearly TREC (Text REtrieval Conference) conferences sponsored by the NIST (National Institute of Standards and Technology, USA). The TREC conferences reflect the need for a more refined automatic indexing of the content of texts as an answer to the shortcomings of current full-text search (see Harman, 1993, 1994, 1995, 1996; Voorhees & Harman, 1997, 1998, 1999).

7.2 Relevance Feedback

An important and difficult operation in information retrieval is generating useful query statements that can extract all the relevant documents wanted by the users and reject the remainder. Because an ideal query representation cannot be generated without knowing a great deal about the composition of the document collection, it is customary to conduct searches iteratively, first operating with a tentative query formulation, and then improving

formulations for subsequent searches based on the evaluations of previously retrieved materials. One method for automatically generating improved query formulations is the well-known *relevance feedback* process.

Methods using relevance information have been studied for decades and are still investigated. Rocchio (1971) was the first to experiment with query modification and with positive results. Ide (1971) extended Rocchio's work. Salton and Buckley (1990) compared this work across different test collections. Relevance feedback is extensively studied in the *Text REtrieval Conferences (TREC).*

The main *assumption behind relevance feedback* is that documents relevant to a particular query resemble each other. This implies that, when a retrieved document has been identified as relevant to a given query, the query formulation can be improved by increasing its similarity to such a previously retrieved relevant item. The reformulated query is expected to retrieve additional relevant items that are similar to the originally identified relevant item. Analogously, by reformulating the query, its similarity with retrieved non-relevant documents can be decreased.

So, *a better query is learned* by judging retrieved documents as relevant or non-relevant. The original query can be altered in two substantial ways (Salton, 1989, p. 307). First, index terms present in previously retrieved documents that have been identified as relevant to the user's query are added to the original query formulation. Second, using the occurrence characteristics of the terms in the previously retrieved relevant and non-relevant documents of the collection allows altering the weight of the original query terms. The weight or importance of query terms occurring in relevant documents is increased. Analogously, terms included in previously retrieved non-relevant documents could be de-emphasized. Both approaches have yielded improved retrieval results (Salton & Buckley, 1990; Harman, 1992b). Experiments indicate that performing multiple iterations of feedback until the user is completely satisfied with the results, is highly desirable.

Relevance feedback is used both in ad-hoc interactive information retrieval and document filtering based on long-term information needs.

Although relevance feedback is considered as being effective in improving retrieval performance, there are still some *obstacles.* One should be selective of which terms to add to the query formulation (Harman, 1992b) and the weights of which terms of the query formulation to alter (Buckley & Salton, 1995). Moreover, current text collections often contain large documents that span several subject areas. It has been shown that trimming large documents by selecting a good passage when selecting index terms, has a positive impact on feedback effectiveness (Allan, 1995).

7.3 Information Agents

There are many definitions of the concept "*agent*" (we refer here to Bradshaw, 1997, p. 3 ff,). A crude definition is that an agent is software that through its imbedded knowledge and/or learned experience can perform a task continuously and with a high degree of autonomy in a particular environment, often inhabited by other agents and processes (cf. Shoham, 1997). There is an emerging interest in the engagement of information agents (Croft, 1987; Standera, 1987, p. 217 ff.; Maes, 1994; Koller & Shoham, 1996). An information agent supplies a user with relevant information that is for instance drawn from a collection of documents.

The main goal of employing an information agent in information selection and retrieval is to determine the user's real need and to assist in satisfying this need. However, there is a growing interest in agents that identify or learn appropriate content attributes of texts.

1. A typical task in an information retrieval environment is *filtering of information* according to a profile of a user or a class of users (Allen, 1990). Such a profile is called a *user's model.* The agent knows the user's interests, goals, habits, preferences and/or background, or gradually becomes more effective as it learns this profile (Maes, 1994; Koller & Shoham, 1996). The knowledge in the profile is *intellectually acquired* (from the user and experts), implemented and maintained by knowledge engineers, Or, the *knowledge* is *learned* by the agent itself based on good positive (and negative) training examples. Learning a user's profile has multiple advantages, including the avoidance of costly implementation and maintenance, and easy adaptations to changing preferences. Learning of users' preferences is closely related to the technique of relevance feedback. Again, such an approach assumes the relevancy of documents that are similar to previously retrieved documents found relevant.
2. Information agents also perform other functions, which support the retrieval operation. They can provide the *services* of a thesaurus, such as providing synonyms to query terms or supplying broader or narrower terms for the query terms (Wellman, Durfee, & Birmingham, 1996; see chapter 5). An agent can also select the best search engine based upon knowledge of search techniques.
3. Research on information agents especially focuses upon the characterization and refinement of the information need. It is equally important to automatically identify or *learn appropriate content attributes of texts* (Maes, 1994). If we obtain a fine-grained and clear user's request, an almost similar fine-grained characterization of the

content of a document is needed for an accurate comparison of information need and document.

7.4 Document Engineering

The technological shift to multimedia environments affects the coding and structure of electronic documents. Electronic documents become more complex, They are bestowed with attributes, which form a *document description.* Also the linguistic text message in an electronic medium is structured and delivered distinctively from the print and paper medium (McArthur, 1987). Texts have stylistic attributes (e.g., used style and fonts), extensional attributes (e.g., name of the author, date of creation), which are also called objective identifiers, and content attributes (e.g., key terms, links), which are called non-objective identifiers (cf. Salton, 1989, p. 276). These attributes are recognizable by their mark-ups in the document. Different *standards* for document description allow using the documents and their attributes independent of the hardware and the application software. Examples of such standards are SGML (Standard Generalized Mark-up Language) and HTML (HyperText Markup Language). The use of such mark-ups greatly benefits the accessibility of the information contained in and attached to the documents.

Despite the appeal and promise of such an approach, one must be aware of its limits among which the complexity and cost of assigning the mark-ups. The creation of current and future electronic documents is sometimes compared with the creation of software (Walker, 1989). Hence, the term *document engineering* is in use. Creation of electronic documents is a complex task. Compared to the field of software engineering there is a clear need for modularity, abstraction, and consistency. Objective identifiers, such as authors' names, publisher's names, and publication date, in general pose no dispute about how to assign them. When mark-ups regard content attributes (e.g., key terms and hypertext links), one must be aware of costly and sometimes subjective and inconsistent attribution of these attributes. The intellectual assignment of content mark-up is considered as a form of manual indexing (Croft et al., 1990). Multiple studies indicate that manual indexing is inconsistent and subjective (Beghtol, 1986; Collantes, 1995). "*Interindexer consistency*" exhibits a direct positive influence upon retrieval effectiveness (paper of Leonard cited in Ellis, Furner, & Willett, 1996). Yet, we don't have many studies about "*interlinker consistency*". A study of Ellis, Furner-Hines, and Willett (1994) shows little similarity between the link-sets inserted by different persons in a set of full-text documents. These authors were not able to prove a positive relationship between inter-linker

consistency and navigational effectiveness in hypertext systems (Ellis et al., 1996). The problem might be alleviated when the text writer acts as a document engineer and is responsible for assignment of content attributes and links. In this way, the writer of text defines possible text uses and navigation between texts (cf. Barrett, 1989; Frants, Shapiro, Voiskunskii, 1997, p. 137). Moreover, the document engineering is not always cost effective, especially when dealing with heterogeneous material such as text content. Because of a better accessibility of the information through document mark-ups, time is gained when searching information. However, extra time is needed to accurately assign mark-ups.

Hence, the document engineer could use some extra *automatic support for assigning content attributes to texts* at the time of document creation (Alschuler, 1989; Wright & Lickorish, 1989; Brown, Foote, Jones, Sparck Jones, & Young, 1995). Especially for large active document collections, such as news texts, intended for a heterogeneous audience, this might be beneficial (Allen, 1990).

8. THE NEED FOR BETTER AUTOMATIC INDEXING AND ABSTRACTING TECHNIQUES

Written as well as spoken text is a very important means of communicating human thoughts and knowledge. In our current information society, we are *overwhelmed with electronic textual documents.* Document collections are constantly growing and their content is constantly evolving. Information retrieval and selection systems are becoming of increasing importance. They must help us to find documents or information relevant to our needs

Written text is considered as an intricate cognitive phenomenon. The cognitive process of creating and *understanding natural language text* is complex and not yet completely understood. However, it is clear that besides coding and decoding linguistic signs, it involves additional cognitive processes. Communication through natural language text is basically *ostensive* and *inferential.* The creator ostensively signals his or her communicative goals. The inferential character of understanding natural language is one of the factors that makes an automated understanding of text a difficult operation. The inferences refer to knowledge that is shared by the text's creator and user and that is not made explicit in the text. The inferences also refer to the individual cognitive state of the user and allow determining the meaning of a text to the individual user.

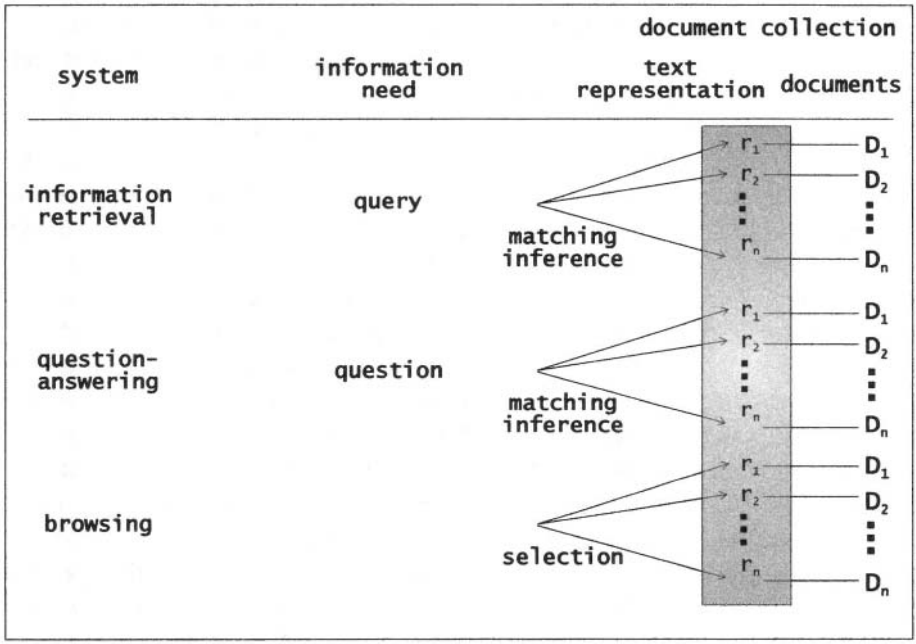


Figure 3. The importance of the text representations ($r_1 \ldots r_n$) in information retrieval and selection.

It is important for the user of a document collection to find documents or information that is relevant for his or her need. Even, if a user has no well-defined information need and wants to browse the document collection, he or she wants to be guided in his or her selection of documents. Information retrieval and filtering systems, question-answering systems, and browsing systems that operate upon textual documents all rely upon characterizations of their content (Figure 3). These *text representations* are the result of *indexing and abstracting the texts.* The text representations are matched with representations of the information need or guide the user in selecting relevant documents or information. The quality of the retrieved and selected information is becoming of increasing importance (Convey, 1992, p. 105). The users of the still expanding electronic databases and libraries want to retrieve all relevant documents or information, but do not want to be overwhelmed with documents that are irrelevant or only marginally relevant

to their needs. The users of browsing systems want to be effectively directed towards interesting documents, without being submerged in possible choices. Currently, this is far from realized for textual databases. There is a real information (retrieval) problem. The problem is caused by incorrect and incomplete representations of an information need and of the content of document texts, and by a probabilistic matching between both.

Indexing commonly extracts from or assigns to the text a set of single words or phrases that function as key terms. Words or phrases of the text are commonly called natural language index terms. When the assigned words or phrases come from a fixed vocabulary, they are called controlled language index terms. The index terms, besides reflecting content, can be used as access points or identifiers of the text in the document collection. This form of text representation is used in information retrieval and filtering systems (Figure 3). *Abstracting* results in a reduced representation of the content of the text. The abstract usually has the form of a continuous, coherent text or of a profile that structures certain information of the text. Abstracts are mainly used in question-answering systems and browsing systems (Figure 3). Indexing and abstracting of the content of texts are traditionally manual tasks. In the growing document collections, the task of human indexing and abstracting is not feasible in terms of efficiency and cost. Moreover, the manual process is not always consistently done. However, current text representations that are automatically generated do not accurately and completely represent the content of texts. Better automatic indexing and abstracting techniques certainly contribute in resolving the information retrieval problem.

Other solutions to the information retrieval problem have been proposed with some success. We saw that full-text search, relevance feedback, information agents, and document engineering all contribute to more effective information retrieval and selection systems. We also demonstrated that each of these answers benefit from a more refined characterization of the content of texts.

Full-text search is the simplest form of automatic indexing. It is generally assumed that inferior results of a full-text search are due to poor automatic identification of good content terms in the texts. *Relevance feedback* will be improved when more selectively content is identified in the documents, which will be used in reformulating the query. Especially, when employing long documents in a feedback process, such a selection is necessary. The development of *information agents* goes hand in hand with the need for a more refined automatic characterization of the content of text. When learning a user's profile, content features need to be identified in the document texts that are salient for the learning of the profile and that permit

comparisons with a detailed profile. *Multimedia information systems* are being developed worldwide, The *content* of each object in a multimedia system (including *textual objects*) needs to be represented. Without such a representation, the system would not be able to integrate information from different media. At present, the representation of textual objects is done by intellectual attribution of key terms that should reflect the content, by intellectually linking text items that treat similar or related contents, or by intellectually creating abstracts that help in document selection. Here again, there is a need for an effective automatic characterization of the texts' contents.

The above considerations all stress the need for more refined procedures for automatically indexing and abstracting texts. This brings us back to the point where we started the reasoning in this chapter. Natural language understanding of text is a difficult task. However, we feel that progress in content understanding is possible without relying upon a complete and complex processing of the texts aiming at their full understanding.

1. Progress can be made in defining the *aboutness* or the topics of a text. Despite considerable improvements, we are still not perfect when automatically identifying the aboutness of a text. Ideally, a text should be represented by different levels of aboutness, allowing for a motivated zooming of its topics and subtopics (Lewis & Sparck Jones, 1996). Aboutness is a permanent quality of the text and has proven in the past its usefulness in information selection. As a cognitive model of text comprehension, the Kintsch and van Dijk model (1978) has a potential for the automatic recognition of the aboutness of a text (Endres-Niggemeyer, 1989; Pinto Molina, 1995).
2. If indexing and abstracting techniques can correctly characterize the detailed topics including specific information in texts, the detailed topics might correspond to a certain need of a user at a specific moment, Presently, the words of the full-text are insufficiently powerful to capture such a detailed content.
3. We need better techniques for extracting content from text that relates to the *meaning* that users may attach to the text (Fidel & Efthimiadis, 1994). This seems a challenging task, but at least we can concentrate on those cases where texts are used with clear goals that are shared among a class of users (cf. Kintsch & Van Dijk, 1978). This refers to what Maron (1977) calls the retrieval aboutness, which is the meaning of a text to a class of users.

Of course, the challenge is to identify a text's content without having to process it based upon a complete linguistic, domain-world, and contextual knowledge of the communication. We think that improvements are possible having a limited amount of knowledge added or having the knowledge automatically acquired. Using a minimum of knowledge sources in text understanding fits in with traditional research in automatic indexing and abstracting in the field of information retrieval. Document collections are often very heterogeneous and are composed of texts of different types and origins. We especially focus upon techniques for better identification and extraction of content terms, indexing of sections or passages, automated methods for assignment of subject codes, information extraction, and text summarization techniques (cf. Carbonell, 1996).

We conclude that there is an absolute need for refined techniques for automatically indexing and abstracting document texts. These techniques form the subject of this book.

---

[1] In this book we make the following distinction between the terms "data", "information" and "knowledge" (cf. Pao, 1987, p. 10-11). Data are sets of symbols representing captured evidence of transactions and events. We use the term information for selected data. When we use the term knowledge, it refers to knowledge acquired by humans when executing a task or to knowledge as implemented in and employed by knowledge-based systems. The term "information retrieval" sometimes refers to information management in general, more often it refers to the retrieval of documents that satisfy a certain information need. The term is used in both Senses in this book.

Chapter 2

THE ATTRIBUTES OF TEXT

1. INTRODUCTION

In this chapter we analyze text and its components in order to define the essential attributes of written text. A substantial number of the attributes described also apply to spoken text. We focus on text written in Western European languages without going into too much detail about the language aspect of text. Illustrations of text attributes refer to text written in English. If illustrations are drawn from Dutch text, they are quoted in Dutch and translated in English.

Written text has three major components. Its *layout structure* concerns extra-textual elements, such as fonts, font styles, and colors. The *logical structure* bears on the organization of chunks of information in, for instance, chapters, paragraphs, and information nodes. The layout and logical structure refer to the presentational structure and are bound to the medium and technology of the communication process. The third component is *text content.* We focus upon attributes of text that relate to its content.

2. THE STUDY OF TEXT

As indicated by its definition (see chapter 1) text is composed of linguistic units, *Linguistics* is the scientific and rigorous study of the formal nature of language (Ellis, 1992, p. 28). The interdisciplinary science of text, also called text *linguistics,* describes and explains shared features and

functions of texts (de Beaugrande & Dressler, 1981, p. 3; van Dijk, 1997). Its task is to describe and explain the mutual relationship of different aspects of the forms of language use and communication in different disciplines (van Dijk, 1978, p. 8). Text linguistics also investigates what standards texts must fulfill and how texts might be produced or received.

Text linguistics is a subfield of the broader interdisciplinary study of *discourse analysis* (de Beaugrande, 1985). Discourse is a form of language use (van Dijk, 1997). The term "discourse" usually refers to spoken as well as to written language use, though sometimes this concept is extended to include other types of semiotic activity (i.e., activity that produces meanings), such as visual images (e.g., photography, film, video, diagrams) and non-verbal communication (e.g., gestures) (Fairclough, 1995, p. 54). Discourses (including texts) usually belong to a specific genre or type (e.g., letter, news story). A genre has a specific structure, i.e., a defined organization of its components (Fairclough, 1995, p. 76). We speak of a text genre or text type in case of textual material. In the case of multi-media documents, we usually speak of a discourse type or genre. Although the term "type" and "genre" are habitually used as synonym, sometimes a distinction is made by defining a discourse type as having the property of being drawn upon two or more genres (Fairclough, 1995, p. 76). In this book we will use the terms "text type" and "text genre" as synonyms. Discourse analysis is also related to *pragmatics* (van Dijk, 1997). Pragmatics is the study of the use of language in the communication context. It describes how sentences are used to convey information or how they make a cognitive state of their creator manifest (Dean, Allen, & Aloimonos, 1995, p. 490).

Discourse analysis describes and explains the *properties of text types.* At a *micro level of description,* discourse analysis concerns the vocabulary, syntax, and semantics of the individual sentences, clauses, and phrases (van Dijk, 1997). At a *macro level of description,* discourse analysis goes beyond the sentence boundary and considers the text a complete grammatical unit. It focuses on the ways that sentences are influenced by surrounding sentences. So, it also includes analysis of textual organization above the sentence, including the ways in which sentences are connected together, and the organization of texts (e.g., the organization of turn taking in interviews, the overall structure of a newspaper article). It has been demonstrated that text at this macro level exhibits several structures. An interesting aspect of discourse analysis describes and explains these text structures. Another aspect studies how "surface" linguistic forms or phenomena signal the text structures and explains why these forms are chosen.

Besides the properties of text, discourse analysis studies the characteristics of the *social situation of the communicative event* that

systematically influence the text, i.e., the context of the text (van Dijk, 1997).

When further describing the attributes of text, we follow the distinction of micro and macro level descriptions proposed by van Dijk (1997). When text is described from the viewpoint of text comprehension, Haberlandt and Graesser (1985) differentiate between a *word*, *sentence*, and *text* level. The word and sentence levels respectively regard lexical encoding and access, and sentence segmentation and interpretation, while the text level bears upon topic identification, knowledge activation, and intersentence integration.

3. AN OVERVIEW OF SOME COMMON TEXT TYPES

There is an enormous diversity of kinds of texts (e.g., road sign, nursery rhyme, textbook, scientific article). But, there is a lack of a verifiable taxonomy of text types (Pinto Molina, 1995). In this section we give an overview of some common written text types without the purpose of being exhaustive.

Texts are often distinguished by their function. A second important distinction is between expository text, narrative text, and text types bound to a specific discipline.

Regarding the *function of text*, Halliday (1989, p. 40 ff.) distinguishes text written in order to undertake action (e.g., public signs, product labels and instructions, recipes, maps, television and radio program guides, bills, menus, telephone directories, ballot papers, computer manuals) or to make social contacts (e.g., letters, e-mails, postcards), text written to provide information (e.g., newspaper and magazine articles, scientific articles and reports, patient reports, political pamphlets, informative books, public notices, advertisements, travel brochures), and texts written for entertainment (e.g., magazine articles, strips, poetry and drama texts, novels, essays, film subtitles).

A distinction is often made between expository and narrative text (Rau, Jacobs, & Zernik, 1989). *Narrative text* focuses on the plot of the story, which consists of several actions. The text is usually constructed in a way the reader can easily follow the actions. Examples of narrative texts are news articles, novels, and short stories. In *expository text* there is more emphasis on the topics and subtopics of the text. Here, the organization of the text is important to efficiently find the information regarding the topics in the text. Scientific texts are an important part of expository texts (e.g., encyclopedic articles, scientific articles, technical documentation).

Beside expository and narrative text, there exist text types that are part of specific disciplines. Often, these disciplines employ their own, distinct types that require specific explanations (van Dijk, 1978, p. 19 ff.).

Legal documents present themselves in rather conventional forms that define several types (Danet, 1985; Gunnarsson, 1997; Moens, Uyttendaele, & Dumortier, 1999b). Some of these texts may be part of statute law (treaties, statutes, royal decrees, ministerial decrees, local decrees, etc.). Their function is to state the general rules that everybody should follow. They are officially published, and all citizens are supposed to be aware of their content. Other texts are related to the judicial proceedings: police statements, warrants, official pleadings, and court decisions. Each of them indicates a certain step of the procedure, and serves as an official proof thereof. A third kind of texts is drawn up as a legal proof in the commercial field, i.e., deeds, contracts and articles of association. Moreover, a number of texts are used for administrative reasons (e.g., tax returns). Finally, there are the texts of legal doctrine made up for scientific or research purposes.

Other fields employ specific text types. In the *medical field*, clinical texts present themselves in different types (e.g., text reports that accompany the results of technical examinations, patient history reports, discharge summaries, mail between practitioners, drug prescriptions, patient referrals). In *politics* there are political comments and party programs. In the *economic field* there are stock market reports, invoices, and contracts. *Religion* handles typical text types such as biblical writing, hymns, and psalms.

For indexing and abstracting purposes, we are especially interested in texts that have an informative function. They are the primary texts that are retrieved from documentary databases. Some other text types with an entertainment function (e.g., magazine articles) are interesting to automatically index, facilitating a consequent automated selection.

4. TEXT DESCRIBED AT A MICRO LEVEL

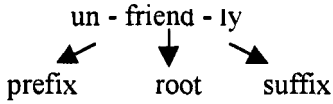
The basic units of text are words. At a more detailed level of analysis text consists of letters, which are the basic symbols of written text, and of phonemes, which are the basic sound units of spoken text. Letters and phonemes separately do not have any meaning, but they combine into small meaning units called morphemes, which form the components of which words are constructed. Words themselves combine into larger meaningful, linguistic units such as phrases, clauses, and sentences. Letters and a number of marks form the character set of electronic texts.

4.1 Phonemes and Letters

The science of phonology analyses the basic sound units of which words are composed. A *phoneme* is the smallest unit of speech that distinguishes one utterance from another, The phoneme is the fundamental theoretical component of the sound system. By borrowing symbols from the Phoenicians, the ancient Greek developed the *alphabet,* a set of symbols (letters) that is the basis of the character set employed in Western European languages. In principle one letter represents one phoneme, which was more or less the case with the ancient Greek alphabet (Halliday, 1989, p. 22 ff.). During history of mankind, languages evolved (dialects and borrowings from other languages), and written language nowadays only approaches the phonetic sounds, and the one to one correspondence between spelling and sounds is often lost. For instance, it is possible for the same letter to represent different sounds. The letters (a-z) can be capitalized (A-Z). Capitalized letters usually have a specific function (Halliday, 1989, p. 33).

4.2 Morphemes

Morphology is the study of the structure of words and describes how words are formed from prefixes, suffixes, and other components. The components of words are called *morphemes*[1] (Ellis, 1992, p. 33 ff. ; Allen, 1995, p. 23; Dean et al., 1995, p. 490). A word consists of a *root* form (*stem* or *base* word) and possibly of additional affixes. For instance, the word "friend" is considered as the root of the adjectives "friendly" and "unfriendly". The adjectives are constructed by adding the *suffix* "ly" to the root and "unfriendly " is constructed by an extra addition of the *prefix* "un". A more complicated construction concerns the derivation of the noun "friendliness" from the adjective form "friendly".



Morphemes are the components of language to which a meaning is associated.[2] The root comprises the essential meaning of a word. The root is a *free morpheme,* because it may occur in isolation and cannot be divided into smaller meaning units. An affix is called a *bound morpheme,* because it must be attached to another meaning unit. There are two classes of bound morphemes. *Inflectional morphemes* do not modify the grammatical category of the base word (e.g., noun) into another category, but signal

changes in, for instance, number, person, gender, and tense. *Derivational morphemes* do modify the category of the base word (e.g., "friendly": the derivation of an adjective from a noun). Morphemes can change forms (e.g., the past-morpheme making "ran" out of "run"). The construction of words from morphemes is rule governed. The rules are language-dependent.

4.3 Words

A *word* is the most basic unit of linguistic structure. A word in written text consists of a string of characters and is delimited by white space or blank characters (possibly in combination with punctuation marks). The words of the text make up the vocabulary of the text.

Words are divided into *categories*, often called *word classes* or *parts-of-speech* (Allen, 1995, p. 23 ff.). This categorization is motivated by the evidence that depending upon its category a word differently contributes to the meaning of a phrase or is a distinct component of a syntactic structure. According to its class, a word may refer to a person or an object, to an action, state, event, situation, or to properties and qualities. For instance, words of the class *noun* identify the basic type of object, concept, or place being discussed, and the class *adjective* contains these words that further qualify the object, concept, or place. Secondly, according to their class, words are specific components of syntactic structures. For instance, an adjective and a noun may be combined into the syntactic structure noun phrase. A word may belong to different categories (e.g., "play" is a noun or a verb).

Some word classes contain words that are better indicators of the content of a text, while other classes contain words that have more strongly pronounced functional properties in the syntactic structures in which they play a role. In this respect a distinction is made between content and function words (Halliday, 1989, p. 63 ff.; Dean et al., 1995, p. 491 ff.). *Content words* serve to identify objects, relationships, properties, actions, and events in the world. Usually, four important classes of content words are considered. Nouns describe classes of objects, events, or substances. Adjectives describe properties of objects. Verbs describe relationships between objects, activities, and occurrences. Here, temporality and aspect of the verb play an important role in shaping the semantic expression of an utterance (Grosz & Sidner, 1986; Dorfmüller-Karpusa, 1988). Adverbs describe properties of relationships or other properties (e.g., "very"). *Function words* serve a more structural role in putting words together to form sentences. They tend to define how content words are to be used in the sentence, and how they relate to each other. They are lexical devices that serve grammatical purposes and

do not refer to objects or concepts of the world. A function word is often small, consisting of only a few letters[3], and its frequency of occurrence in text is usually much higher than the frequency of occurrence of a content word.[4] Function words belong to syntactic classes such as articles, pronouns, particles, and prepositions. The following four classes of function words are often distinguished. Determiners indicate that a specific object is being identified (e.g., "a", "that"). Quantifiers indicate how many of a set of objects are being identified (e.g., "many"). Prepositions signal a specific relationship between phrases (e.g., "through"). Connectives indicate the relationships between sentences and phrases ("and", "but").

A word has a meaning or sense, which is known as *lexical meaning* (Ellis, 1992, p. 38). The lexical meaning or *semantics* of words concerns what words symbolize including their denotations and connotations. The origins and usage of words in certain text contexts define lexical meaning. Dictionaries document the different meanings of words. The meaning of a word in a text is not always clear-cut, as it is illustrated by the following.

1. A word can have more than one meaning (e.g., the word "sentence" can refer to a text sentence, to a court sentence, and to the act of sentencing). The multiple meanings of one word are known as homonymy and polysemy (Krovetz & Croft, 1992). In written text a *homonym* is a word that is spelt in the same way (i.e., *homograph*) as another word with an unrelated meaning. Homonyms are derived from different original words. One speaks of *polysemy* when a word has different, related meanings. The "bark of a dog" versus the "bark of a tree" is an example of homonymy; "opening a door" versus "opening a book" is an example of polysemy. Sometimes, a word does not only belong to different word classes, each of them pointing to a group of possible word senses, within the word class the word can still have different distinct meanings. When words with multiple meanings occur in phrases or sentences, they often only have a single sense, since the words of the phrase or sentence mutually constrain each other's possible interpretations.
2. Different words can have the same meaning (e.g., "vermin" and "pests"), which is known as *synonymy*. Often, different words or phrasal combinations of words express the same concept. Near synonyms are words with a closely related meaning (e.g., "information" and "data"). Also, one word can generalize or specify the meaning of the other word (e.g., the word "apple" specifies the word "fruit").
3. An author has great freedom in the choice of words and can even make up new words or change the meaning of familiar ones. So, a word or a combination of words can be used metaphorically and have a figurative

interpretation in order to create an aesthetic, rhetorical or emotive effect (Scholz, 1988). The use of *metaphors* is almost unlimited. No dictionary can answer all figurative uses of a word or of a combination of words.

4. A word can refer to another word in the text for interpretation. *Anaphora*[5] are textual elements that refer to other textual elements with more fully descriptive phrasing found earlier in the text (called correlates) and that share the meaning of the correlates (Halliday & Hasan, 1976, p. 14 ff.; Liddy, 1990) (e.g., the words "it " and "his" in "The student buys the book and gives it to his sister."). Anaphora are used quite naturally and frequently in both written and oral communication to avoid excessive repetition of terms and to improve the cohesiveness of the text. A *cataphoric reference* is a word that refers to another word further in the text (Halliday & Hasan, 1976, p. 56 ff.). It is also worth mentioning that a word or some words can be omitted from the text. This is called *ellipsis* (e.g., "David hit the ball, and the ball (hit) me.") (Allen, 1995, p. 449 ff.).

4.4 Phrases

Words combine into phrases. A *phrase* consists of a head and optional remaining words that specify the head word (Halliday, 1989, p. 69 ff.; Dean et al., 1995, p. 492 ff.). The *head* of the phrase indicates the type of thing, activity, or quality that the phrase describes. The remaining words are called *prehead modifiers*, *posthead modifiers*, and *complements* according to their location in the phrase. Modifiers and complements can form phrases themselves. Complements are the phrases that immediately follow the head word. For instance, the phrase "this picture of Peter over here" consists of a prehead modifier "this", a head "picture", a complement "of Peter" and a posthead modifier "over here".

The four classes of content words provide the head words of four broad classes of phrases: noun phrases, adjective phrases, verb phrases, and adverbial phrases (Allen, 1995, p. 24 ff.). A fifth class of phrases is built with a preposition and a noun phrase.

1. *Noun phrases* are used to refer to concepts such as objects, places, qualities, and persons. The simplest noun phrase consists of a single pronoun (e.g., "she", and "me"). A proper name forms another basic noun phrase, consisting of one or multiple words that appear in capitalized form in many Western European languages (e.g., Los Angeles). The remaining forms of noun phrases consist of a head word, and possibly of other words that qualify or specify the head.

2. An *adjective phrase* can be a component of a noun phrase, when it modifies the head noun. It also occurs as the complement of certain verbs (e.g., “heavy” in “it looks heavy”). More complex forms of adjective phrases include qualifiers preceding the adjective (e.g., “terribly” in “terribly dangerous”) as well as complements after the adjective (“to drive” in “dangerous to drive”).
3. A *verb group* consists of a head verb plus optional auxiliary verbs. Auxiliary verbs and the forms of the head verb combine in certain ways to form different tenses, aspects, and active and passive forms (e.g., “has walked”, “is walking”, and “was seen”). Some verb forms are constructed from a verb and an additional word called a particle (e.g., “out” in “look out”). A *verb phrase* consists of a verb form and optional modifiers and complements. Verb phrases can become quite elaborated consisting of several composing phrases (e.g., “gave the sentence to the accused without hesitation”).
4. An *adverb phrase* consists of a head adverb and possible modifiers (e.g., “too quickly”).
5. The term *prepositional phrase* is used for a preposition followed by a noun phrase, which is called the object of the preposition (e.g., “from the court”). Also other forms of prepositional phrases are possibly (e.g., “out of jail”). Prepositional phrases are often used as complements and modifiers of verb phrases.

Phrases usually form the components of which sentences are built. Isolated phrases (e.g., noun phrases) can be found, for instance, in the headings, subheadings, and figure captions of texts.

Phrases are less ambiguous in meaning than the individual words of which they are composed. But, this is no general rule.

4.5 Sentences

Sentences are used to assert, query, command, or bring about some partial description of the world. The sentence is organized in such a way as to minimize the communicative effort of the user of the text. A sentence is composed of a *topic* and of additions (comment) to the topic (e.g., properties of the topic, relationships with other items, modifications of the topic) (Halicová & Sgall, 1988; Tomlin, Forrest, Pu, & Kim, 1997). The additions to the topic are often called the *focus* of the sentence. For instance, in very simple English sentences the topic is identical with the subject of the sentence and the focus with the predicate. The terms "*theme"* and "*rheme"* are often used as a synonym for respectively topic and focus (Halicová &

Sgall, 1988).[6] The theme is the starting point of the utterance, the object or person about which or whom something will be communicated and the rheme is the new information predicated of the theme (Halliday, 1976; Fries, 1994). The transitional element between theme and rheme is usually a verb, carrying some new information in the sentence, but to a lower degree than the rheme.

An utterance is structurally composed of a topic and a comment of that topic. This structure is closely connected to the way we orally communicate (Halicová & Sgall, 1988). This "deep" structure is considered as being common to all languages. It is coded into a sentence according to the grammatical rules of the language used (cf. Chomsky, 1975; Ellis, 1992, p. 36). A sentence has a *syntactic structure* (Dean et al., 1995, p. 490). It is composed of constituents (phrase classes) that combine in regular ways. In its turn, a phrase is composed of word classes that also combine in regular ways. So, any sentence can be decomposed or altered by the application of certain rules. The structures allowable in the language are formally specified by a *grammar*. The grammar allows decomposing a sentence into phrases of a specific class, which in their turn can be decomposed into words of a specific class. For instance, the sentence (S) "The judge buried the case" consists of an initial noun phrase (NP) and a verb phrase (VP). The noun phrase is made of an article (ART) "The" and a common noun (NOUN) "judge". The verb phrase is composed of a verb (VERB) "buried" and a noun phrase (NP), which contains an article (ART) "the" and a common noun (NOUN) "case". We can define the following set of rules that define the syntactic structure.

<S> ::= <NP> <VP>
<NP> ::= <ART> <NOUN>
<VP> ::= <VERB> <NP>

Based on the grammatical rules, we can produce an unlimited number of sentences and any sentence can be modified and lengthened by adding an infinite number of adjectives and relative clauses (cf. Chomsky, 1975).

The representation of the content of a sentence is called a *proposition* (Allen, 1995, p. 234). A proposition is formed from a predicate followed by an appropriate number of terms to serve as its arguments. "The judge buried the case" can be represented by the proposition (BURY JUDGE CASE). In this proposition the verb BURY has two arguments JUDGE and CASE.

Sentences are usually less ambiguous in meaning than the phrases and individual words they are composed of. The lexical ambiguity of individual

words is often resolved by considering the meaning of the other sentence constituents. Besides unresolved lexical ambiguity, ambiguity in sentence meaning possibly is the result of structural ambiguity (cf. Ellis, 1992, p. 38), when the syntactic structure of a sentence that contributes to the meaning of the sentence is ambiguous (e.g., the sentence "I saw the man with the binoculars"). The meaning of an ambiguous sentence can be disambiguated when considering the meaning of surrounding text sentences.

4.6 Clauses

Complex sentences can be built from smaller sentences by allowing one sentence to include another as *subclause* (Allen, 1995, p. 31 ff.). Common used forms are embedded sentences as noun phrases (e.g., "To go to jail ...") and relative clauses of noun phrases ("... who sentenced the man"). The former form involves slight modifications to the sentence structure to mark the phrase as noun phrase, but otherwise the phrase is identical to a sentence. The latter form is often introduced by a relative pronoun (e.g., "who", "that"). A relative clause has the same structure as a regular sentence except that one noun phrase (e.g., in subject position, object position, object to a preposition) is missing.

Regarding the topic structure, main clauses generally foreground topics, whereas subordinate clauses generally background them.

4.7 Marks

The uses of special symbols that mark up written text have been developed throughout the centuries (Halliday, 1989, p. 32 ff.). The *marks* or symbols help the user of the text to correctly analyze the text. They have three kinds of functions. A first function is boundary marking. For instance, punctuation marks are used to mark off sentences or clauses. Another example is the blank character that delimits words and that is used in texts electronically stored. A second function is status marking indicating a speech function. For instance, an interrogation mark refers to a question, and quotation marks refer to quoted speech. A third function is relation marking. Special symbols indicate linkages, interpolations, and omissions (e.g., hyphen, parenthesis, apostrophe). Besides these special symbols, current texts contain characters that code specific concepts, such as the dollar, the percentage character, and digit characters to write numbers in their digital form.

So, the character set of written texts includes, besides letters and digits, a number of punctuation marks and special characters (e.g., ',', '+', '%'), and

several white space or blank characters in texts electronically stored (e.g., as word delimiters) (cf. Lebart, Salem, & Berry, 1997, p. 37). Although we do not discuss layout characteristics, special layout characteristics (e.g. the use of underlining, characters in larger font, italic and bold character forms) can stress some words or phrases of the text.

5. TEXT DESCRIBED AT A MACRO LEVEL

A text is not simply composed of words, phrases, and sentences, but the sentences and phrases are ordered according to some conventions. Text as a whole has its own syntax and semantics and is characterized by several structures. Text structures are an essential characteristic of written and spoken text and guaranty a text's coherence (Meyer, 1985). Coherence is described in chapter 1 as one of the major characteristics of text and regards the global organization of the discourse (De Beaugrande & Dressler, 1981, p. 84 ff.; Rudolph, 1988). Coherence is to be seen as the connection in the mental representation attributed to the text. Cohesion, which is another important characteristic of text, regards the surface organizational patterns that connect the elements of a text into a whole (De Beaugrande & Dressler, 1981, p. 48 ff.; Rudolph, 1988). The structures and their signaling linguistic cohesive cues (Table 1) are important means for the creator of text to ensure that a user can establish a correct interpretation.

The literature on text structures is very heterogeneous. More studies of text syntax and semantics are needed, providing a description of the properties and organization of different genres of text and providing descriptions across different text types. The following sections attempt to synthesize the main findings in the literature (cf. Moens et al., 1999b).

5.1 The Schematic Structure or Superstructure

Definition

The most typical characteristic of a text type is its overall formal structure, also called *schematic structure* or *superstructure* (van Dijk, 1997). The superstructure of a text type is a conventional (and therefore culturally variable) production scheme to which a text is adapted. The definition of a text type often relies upon its schematic structure. The schematic structure of a particular text type is specified in terms of the ordered parts it is built of.

Table 1. Macro level of text description: Text structures and their main signaling cues.

Text structures	Signaling cues
Schematic structure	Ordering of text segments
	Cue phrases
Rhetorical structure (here intersentential discourse relations)	Ordering of text segments
	Cue phrases
	Pronoun and reference use
	Tense and aspect of verbs
	Marks
Thematic structure	Locational cues
	Cue phrases
	Content terms

The segments are either all obligatory, or some obligatory and some optional. They occur in a fixed or partially fixed order. The segments are combined to create larger parts and whole texts. So, the schematic structure is often hierarchically organized, but segments can also be sequentially organized (cf. Paice, 1991). A text segment can be of different size. It may consist of one sentence or paragraph, span over several sentences or paragraphs, or just be one text statement. Text schemata show the routine and formulaic nature of much text output. The experiments of Dillon (1991) clearly demonstrate that readers who are experienced in reading certain text types possess a superstructure or model of the text that enables them to predict with high levels of accuracy where specific information is located (cf. Reichman, 1985, p. 19). So, the creators and users of these texts (unconsciously) know the text schemata.

Examples

A simple example in the class of *expository text* is the schematic structure of *scientific articles* in the Western culture. A scientific article usually contains the following ordered text segments: purpose of the research, methodology, results, discussion of the results, and conclusions (Pinto Molina, 1995). On a more detailed level of analysis, the schematic structure of scientific articles possibly exhibits variants that are typical of the natural sciences or of the social sciences and humanities.

Text schemata have been extensively studied in case of written *news stories* (van Dijk, 1985, 1988a, 1988b; Bell, 1991). News stories belong to the class of *narrative text.* For instance, van Dijk (1988b) studied the schematic structure of 700 stories from 138 selected newspapers in 99 countries. It was found that the news discourse follows a number of conventional schemata, consisting of categories that are typical for news

discourse. Van Dijk and Bell suggest that a news report has a headline and a lead, which summarize the story, an attribution, which sets the context of the story, an events element, which covers the main events of the story, and a comment element. The schemata of news stones alert us that the immense diversity of events in the world is reduced to often-rigid formats.

Another example of a text type in a *specialized field* (legal field) is the text of a Belgian *correctional case* (Moens & Uyttendaele, 1997), which is composed of the following ordered segments: a superscription, which may contain the name of the court and the date; the identification of the victim; the identification of the accused; the alleged offenses, which describe the crimes and factual evidence; the transition formulation, which marks the transition to the grounds of the case; the opinion of the court, which contain the arguments of the court to support its decision; the legal foundations, which contain the statutory provisions applied by the court; the verdict; the conclusion, which may again contain the name of the court and the date. Some of the composing segments are optional.

Signaling linguistic cues

The schematic structure or superstructure may, but not necessarily, be signaled in the text by surface linguistic forms, such as the use of typical phrases and other lexical cues (Allen, 1995, p. 504 ff.). The explicit use of certain words and phrases are among the primary indicators of text segment boundaries or categories. For instance the beginning of the text segment "transition" of a Belgian criminal case is signaled by the phrase "Gezien de stukken van het onderzoek" ("*Given the documents in the case*").

The schematic structure of a text may, but not necessarily, coincide with the *logical structure* of the document text, which is its presentation structure (e.g., chapters, sections, and paragraphs) (cf. Paice, 1991).

Sometimes, there is no overt linguistic or presentational marker of a segment limit. Then, its limit can be inferred from a relationship with another segment (e.g., preceding or following another segment).

5.2 The Rhetorical Structure

Definition

The term "*rhetorical structure*" finds its origin in Rhetorical Structure Theory (RST), which describes what parts or segments texts have and what principles of combination can be found to combine parts into entire texts (Mann, Matthiessen, & Thompson, 1992). The term rhetorical structure

covers a broad meaning. First, the rhetorical structure refers to the superstructure or schema by which the text type is characterized (see above). Second, it often refers to the structure expressing the organization of coherent, continuous text and to the rhetorical relations that hold between text sentences and clauses, called intersentential discourse relations (cf. Hobbs, 1979; Reichman, 1985, p. 21 ff.).[7] These relations can be simple (e.g., succession, conditionality) or can be semantically more complex (e.g., motivation, circumstance, contrast). In this sense the ultimate aim of Rhetorical Structure Theory is to define a set of domain-independent relationships between sentences that define coherent discourse. Taxonomies of discourse segment relations have been built (e.g., Mann et al., 1992; Hovy, 1993b). Rhetorical relations are applicable to many kinds of texts, enabling a unified description of text structure regardless of text type or genre. It is in this specific meaning that we will use the term rhetorical structure in this book. But, in its broad sense the rhetorical structure specifies the genuinely genre-specific aspects of text structure (superstructure) and the more genre-independent structural aspects.

Examples

In the sentence “The most extreme case of fear I have ever witnessed was a few summers ago when I visited Alaska.”, the subclause “when I visited Alaska” has a rhetorical relation of *circumstance* with the two foregoing clauses of the sentence. The sentence “Fill out the form to become a candidate." demonstrates a relation of purpose. Becoming a candidate presents a situation to be realized by the activity of filling out a form. Another example is formed by the sentences: “A well-groomed car reflects its owner. The car you drive says a lot about you.”. The second sentence is a *restatement* of the first.

Signaling linguistic cues

Creators of text often use specific linguistic signals that indicate rhetorical relations between text sentences and other clauses. Linguistic surface phenomena that signal rhetorical relations are lexical cues, pronoun and other reference use, tense and aspect (Hovy, 1993b). Although we discussed text marks as micro level attributes, some of them may cause a rhetorical relation between sentences (e.g. a question mark will induce an answer in following sentences). The most prominent rhetorical cues are the lexical cues (Allen, 1995, p. 504 ff.), which are also called cohesive elements or devices.

The main function of the *cohesive text elements* is to indicate that there is a rhetorical relationship between the text segments involved that guides the user of the text towards the correct interpretation of the text, For instance, a purpose is detected by the use of the words "in order to" in the sentence "I work hard in order to buy a house." Among the constituting elements of text cohesion, we often find connective expressions and conjunctions, called *connectives* (e.g., "and", "because") (Rudolph, 1988). However, an overt linguistic marker of a rhetorical relation is sometimes missing, making an identification of the rhetorical relation more complicated. For instance, in the above example there is no overt linguistic marker that indicates that the sentence "The car you drive says a lot about you." is a restatement of the sentence "A well-groomed car reflects its owner.". It is also possible that cue phrases function ambiguously with respect to a certain discourse role (Grosz & Sidner, 1986).

5.3 The Thematic Structure

Definition

The *thematic structure* of a text concerns its overall organization in terms of themes or topics. It is usually a hierarchical organization, in the sense that we can identify the theme of the whole text, which can be typically spelt out in terms of a few rather less general themes, which can each in their turn be spelt out in terms of even more specific themes.

Discourse topics represent the aboutness of a text and also its global meaning (van Dijk, 1997; Bánréti, 1981; Halicová & Sgall, 1988; Tomlin et al., 1997). They represent the gist of the discourse, its most important information. The discourse topic(s) of text summarize and categorize the semantic information of the text. The global text topic is the underlying proposition of the text as a global entity, i.e., the kernel representation of its content. The subtopics summarize the more detailed meanings of the discourse that its users possibly assign to the text. Because identifying a text's thematic structure concerns a macro level of analysis (i.e., concerning the overall discourse), sometimes the term "*macrostructure*" is used as a synonym for this structure (van Dijk, 1988b, p. 30 ff.; van Dijk, 1997).

The hierarchical organization of topics and subtopics may, but not necessary, be reflected by a hierarchical organization of topic segments in the text. Other organizations are possible and the thematic organization is often text type dependent (cf. García-Berrio & Albaladejo Mayordomo, 1988). Text segments can have topics of their own (e.g., the topic of a text passage). During the discourse, a topic can be suspended at one point and

later resumed as though it had not been interrupted, which is called a *semantic return* (Allen, 1995, p. 532). Discourse topicality is actually more complex than sentence topicality because it is more difficult to recognize and does require more organizational work than, for instance, the subject of a sentence (Ellis, 1992, p. 119).

At a more detailed level of analysis[8], the topics of sentences or clauses exhibit rhetorical relations with the topics of previous or following sentences or clauses (e.g., contrast, illustration) (van Dijk, 1997). Other forms of *thematic progression* in sentences and clauses are possible: theme repetition (the theme of one sentence is repeated in successive sentences), thematization of the rheme (the rheme of a sentence becomes the theme of the next sentence), topic shifts, and more complex theme progression patterns (Scinto, 1983).

Examples

It is assumed that the main topic of *scientific articles* is discussed throughout the entire text, while the discussion of a subtopic is restricted to the sentences of a passage of the text (Hearst & Plaunt, 1993). The thematic structure of written *news stories* has been studied by van Dijk (1985, 1988a, 1988b) and Fairclough (1995, p. 30). In the news story, the more general topics come first, while the more detailed topics occur further in the story. The last example concerns a text type in a *specialized field* (legal field). In Belgian *criminal cases* the text segment regarding the argumentation of the judge discusses the different crime topics. In this discussion a crime topic can be abandoned and resumed further in the text (Moens, Uyttendaele, & Dumortier, 1999a).

Signaling linguistic cues

The topics of a text are closely related with the surface linguistic phenomena of the text. The creator of a text explicitly *signals* topicality in order to achieve a correct interpretation of the text by its users (van Dijk, 1988b, p. 32 ff.). Research has demonstrated that language users are competent at identifying a text's topics and their boundaries (Ellis, 1992, p. 127), which confirms the presence of *surface cues.* Markers of topicality are more studied in speech than in written text (cf. Ellis, 1992, p. 137). However, it is possible to distinguish a few linguistic phenomena that are helpful in identifying topics and topic boundaries in written text.

1. The *schematic structure orders the thematic content of a text* (Kieras, 1985). For instance, it has been demonstrated that the thematic structure of news stories parallels the news schemata (van Dijk, 1985; cf. Fairclough, 1995, p. 30). The headline of a news report formulates the overall theme of the text. The lead and the attribution contain the most important topics, while subtopics appear in the body of the story.
2. *Locations*, other than the ones defined by the schematic structure, are important for topic identification. The thematic structure is sometimes cued by the logical (presentational) structure of a written text. Thematic units possibly coincide with chapters and paragraphs. The paragraph is often considered as the most identifiable demarcation of a topic (García-Berrio & Albaladejo Mayordomo, 1988; Ellis, 1992, p. 133). Also, topical information is highly present in the first sentence of a paragraph and to some extent to the end of a paragraph (Kieras, 1985). The position of a term within a sentence is also significant (Kieras, 1985; cf. Sidner, 1983).
3. The topics of a text are actually described by the words in the sentences of the text. The use of *content words* and their *frequency of occurrence* in the text are considered as general clues to their topical importance (Salton, 1989, p. 279). Also, references to a particular concept that occur in close proximity of another in the text are good indicators of topicality (Hearst & Plaunt, 1993).
4. There are other surface linguistic cues such as the use of *cue words and phrases* (Kieras, 1985; Ellis, 1992, p. 131 ff.). Examples of such topic indicators are the cue phrases “about” and “speaking of” followed by the topic. Other words cue new topics or topic shifts (e.g., the word “now”).

It is agreed upon that topic recognition and progression in texts are subjects that require further investigation (Hahn, 1990; Hovy, 1993a). The results of this research are especially valuable for automatic indexing and abstracting of texts.

5.4 The Communicative Goal

Definition

The discourse as a whole and its composing segments have an associated purpose. The discourse purpose is the intention or *communicative goal* that underlies engaging in the particular discourse. This intention provides both the reason a discourse is being performed and the reason the particular content of this discourse is being conveyed rather than some other

information (Grosz & Sidner, 1986). Like any form of discourse, a written text has its communicative intention. The communicative goal of a text is often composed of different subgoals (Figure 1). So, it is possible to define a *communicative goal structure*, sometimes called an *illocutionary structure* (Branting, Lester, & Callaway, 1997; cf. Allen, 1995, p. 567) or intentional structure (Grosz & Sidner, 1986).[9] Creators of text use the propositional content of utterances to signal illocutionary acts. Illocutionary acts are utterances that have social and communicative purposes. A text's user must not only understand the words and the syntactical relationships of a text in order to understand its aboutness, but he or she must understand how an utterance is functioning (Ellis, 1992, p. 89 ff.). For instance, if a sign says "Attack Dog on Premises", the aboutness or topic of the sign regards an attack dog. But, the text is also a warning about how you should behave, not a simple statement describing the nature of the animal nearby. The communicative goals are very prominent in informative texts.

In a successful discourse, the contents of the complete text and its composing segments achieve its communicative goal. Each text segment is a step in a plan to achieve the overall communicative purpose of the discourse (Hovy, 1993a). A user accesses the text with a specific *focus of attention* (cf. "attentional state" in Grosz & Sidner, 1986), which according to the task of using the text, may only be part of the creator's communicative goal structure (Figure 1).

Example

The main communicative goal of the legal text of a show-cause order is to establish the prerequisites for dismissal of an appeal (Branting et al., 1997). The main subgoals are:

- -to establish the existence of a jurisdictional defect:
- -establishing the orders being appealed
- -establishing that the notice of appeal was untimely as to one of the orders:
    - -establishing the commencement date of the time for filing a notice of appeal
    - -establishing the due date of the notice of appeal
    - -establishing the actual filing date
    - -ruling that the actual filing date was after the due date
- -to order an appropriate sanction:
    - -ordering a time limit for response
    - -a sanction
    - -a rationale for the sanction.

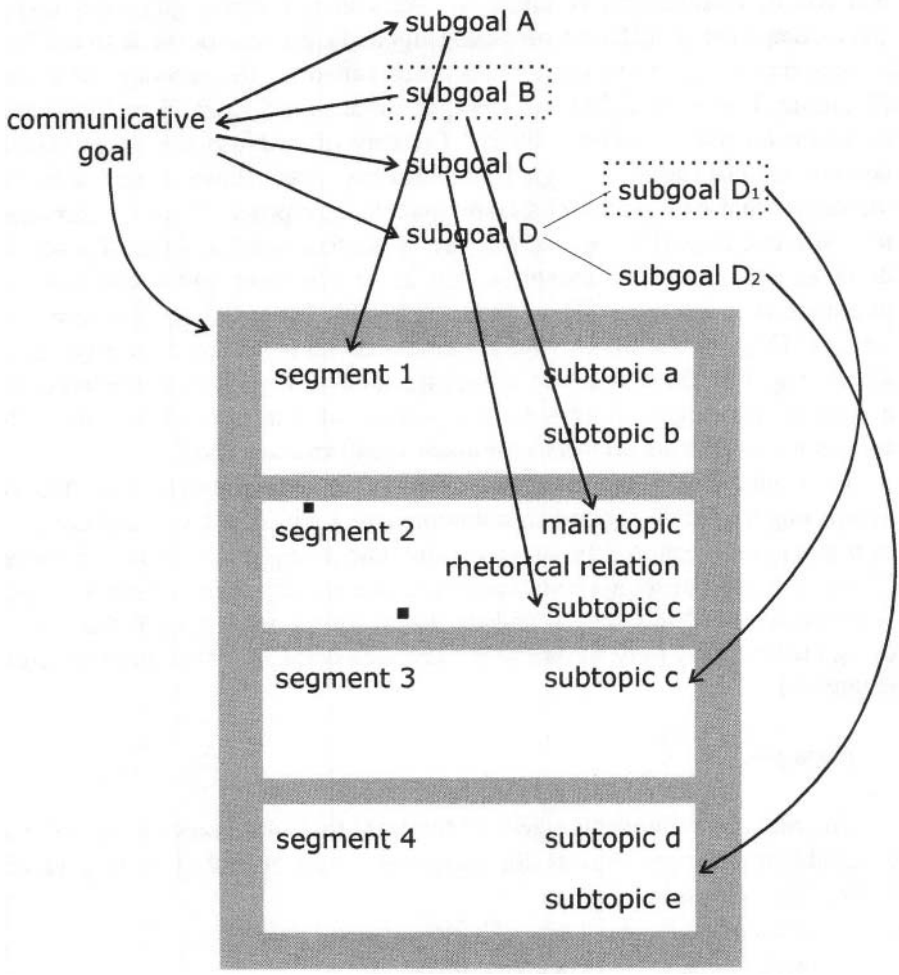


Figure 1. Example of the communicative goals of the creator of the text, the focus of attention of its user, and the relations of the goals with the macro level discourse structures.

Realization of the communicative goal in the text

The communicative goal and the subgoals of a text are realized through the lexical and grammatical expressions in the sentences but also by the text structure (Figure 1). The superstructure, rhetorical structure, and thematic

structure of a text help to realize the communicative goal structure (Hovy, 1993a; Fries, 1994). These structures, which order text content, contribute to the successful realization of the communicative intention. Especially, the superstructure and the rhetorical structure are often closely linked with the communicative goal structure. Without an understanding of the discourse structures by the creator and user of a text, communication is unlikely to succeed (Hovy, 1993b).

It seems that the actual communication through text consists of many *deviations from the ideal structures.*. We find violations of the normative rules for appropriate discourse. According to van Dijk (1997) it is interesting to study these deviations in their own right. Indeed, what appears as a violation of some rule or regularity, may turn out to have a very contextual function. Related to these deviations is the *concept of style* (van Dijk, 1997). Creators of discourse handle different styles. Style is a context bound variation (context regards speaker, perspective, audience, group, etc.) of the expression level of discourse. The concept of style usually assumes that the same concepts can be expressed differently depending on a different communication context. For instance, the choice of a specific word depends upon the audience focused.

5.5 Text Length

The *length* of a text is somewhat dictated by the text type, but this is no general rule. For some text types (e.g., a narrative story), the creator has the freedom to decide himself how many words he or she will use to communicate his or her message. The text length can be computed in different ways. It is usually computed as the number of words or as the number of (different) content words contained in the text (cf. Salton & Buckley, 1988).

6. CONCLUSIONS

Our communication by means of written (and spoken) text is governed by many patterns on a micro as well as on a macro level. Discourses, among which texts, have important communicative goals and subgoals. These intentions are realized with the help of a number of discourse rules that are socially shared by the members of a group, community, or culture. To realize its communicative goals, text exhibits a number of internal structures that go beyond the structure of individual sentences. It is interesting to describe and explain the superstructure, the rhetorical structure, and the thematic structure

of a text as they all contribute to the successful realization of the communicative intention. It is also interesting to see what surface linguistic forms or phenomena signal the text structures. The discourse patterns and rules help govern the selection and ordering of elements in a discourse and make our seemingly randomly organized texts understandable to one another.

In the previous chapter, it was argued that in spite of decades of work on natural language processing, computers are not capable in explaining natural language text in a way done by humans. However, *discourse* studies yield valuable *knowledge* for automatically locating information and content in texts. Discourse patterns that help in identifying the topics of a text are especially interesting. This knowledge can be incorporated in a variety of applications addressing information extraction, such as text indexing and abstracting.

---

[1] In spoken language, phonemes are grouped into syllables. Each syllable is marked by a maximum of acoustic energy in the speech signal. Syllables are produced at a rate of four or five a second in all languages. Most morphemes correspond to single syllables, but there are many that are represented by polysyllabic words.

[2] Phonemes and their corresponding letters carry no meaning, although they discriminate between meanings, as the *n* discriminates between "bar" and "barn".

[3] The "principle of least effort" on the part of the speaker or writer accounts for the fact that the most frequent words tend to be short function words, of which cost of usage is small (Salton & McGill, 1983, p. 60).

[4] Cf. Halliday (1989, p. 64): the *lexical density* (proportion of content words upon total number of text words) may vary according to the type of text.

[5] Anaphoric and cataphoric references are considered as macro level text phenomena, because they safeguard the cohesion of a text between sentences.

[6] About the compatibility of these terms see Tomlin et al. (1997).

[7] Van Dijk (1997) considers the structure expressing intersentential discourse relations as a micro level description of text. We prefer to classify the rhetorical structure as a macro level description, because the rhetorical structure concerns the global organization of the text and a rhetorical relation often connects several sentences in the discourse.

[8] This may be considered as a micro level description of text (cf. van Dijk, 1997).

[9] The communicative goal is not identical to the meaning of a text, but both concepts are related. The communicative goal is a property of the text from the viewpoint of the creator of the text. The meaning is a property of the text from the viewpoint of its user. In successful discourse meaning coincides with the communicative purpose (cf. Hovy, 1993a), but the users of the text are always free to attach additional meanings to it.

Chapter 3

TEXT REPRESENTATIONS AND THEIR USE

1. INTRODUCTION

In the previous chapter we discussed the characteristics of input source texts. Here, we will elaborate on the output of the indexing and abstracting process. We will refine some of the concepts mentioned in the preface and define related concepts. The result of indexing or abstracting the content of text is a text representation. Different forms of text representations are discussed. Before describing the automatic methods in part II of this book, it seems useful to outline the intellectual process of indexing and abstracting. Furthermore, the use of text representations in text browsing, retrieval, and interrogation is important to understand their current form. The storage of the indexing and abstracting products is beyond the subject of this book and is only very briefly referred to. Finally, the main characteristics of valid text representations are given.

2. DEFINITIONS

Text indexing and *abstracting* are processes that create a short description or characterization of the content of the original text (Rowley, 1988, p. 48; Salton & McGill, 1983, p. 52; Lancaster & Warner, 1993, p. 79 ff.). The result of these processes is a *representation* or *representative* of the text, which has a recognized and accepted style or format. Indexing commonly assigns to or extracts from the text a set of words and phrases. Besides

reflecting content, the *index terms* can be used as access points or identifiers of the text, by which the text can be located and retrieved in a document collection. Abstracting generates a summary of the text's content, which has various possible formats. Text indexing and abstracting refer to the human intellectual process as well as to the automated process. Indexing sometimes refers to the automated process of storing text representations in data structures (such as inverted files) to assure an efficient access to the documents that they represent. In this book, we will not use the term indexing in this sense.

Both terms "representation" and "representative" are used for naming the condensed characterization of the content. We prefer using the term *text representation* throughout this book. This term is also used for intermediate representations that are made of text during its indexing or abstracting (cf. Lancaster 1991, p. 219 ff.; Lancaster & Warner, 1993, p. 243). Also, the term *document representative* may refer to the product of indexing and abstracting text (van Rijsbergen, 1979, p. 14; Lewis, Croft, & Bhandaru, 1989). We feel that this term is too general, because it can also refer to content descriptors of other media than text (e.g., images) in a multi-media context or to context descriptors of a document (called objective identifiers) such as the date of creation and name of the author.

A representation is made from the complete text or from certain text passages, the latter being referred to as *passage indexing* (Salton, Allan, & Buckley, 1993). Text representations are used in many forms. The most common are *natural language index terms* identified in the texts and *controlled language index terms* assigned to the texts. The terms form the *indexing language* (Cleveland & Cleveland, 1990, p. 78; Rowley, 1988, p. 52). *Abstracts* usually have the form of text profiles that structure certain information of the text or of continuous, coherent text. They describe a text's content in a more detailed and structured way than index terms.

3. REPRESENTATIONS THAT CHARACTERIZE THE CONTENT OF TEXT

3.1 Set of Natural Language Index Terms

Indexing often consists of drawing natural language index terms directly from the document text (Lancaster & Warner, 1993, p. 80 ff.). This process is called *extraction indexing.* The index terms extracted are content terms in the form of single words or phrases (Harter, 1986, p. 42). The number of

terms extracted varies from a few ones to a large number, depending upon the need to more or less in detail represent the text's content (Salton, 1975b, p. 17). The index terms can have a weight indicating their importance in representing content (Sparck Jones, 1973). Full-text search (see chapter 1) is the simplest form of extraction indexing: Each word in the text can act as an index term.

Indexing with natural language terms has advantages and disadvantages (Blair & Maron, 1985; Harter, 1986, p. 51 ff.; Lancaster, 1986, p. 161 ff.; Furnas, Landauer, Gomez, & Dumais, 1987; Salton, 1989, p. 276; Krovetz & Croft, 1992). It has the *advantage* of being very expressive and flexible, of representing a variety of access points and perspectives of a text, and of easily representing new and complex concepts. The indexing vocabulary is less tightly controlled than controlled language index terms and a great variety of index descriptors is normally identifiable. Because of the lack of fixed index terms, the natural language index terms make a textual database portable and compatible across different document collections. There are however *disadvantages.* The words of the text have the property of being potentially ambiguous (e.g., homonyms). Index phrases are usually less ambiguous because each content word in the phrase provides the context for the others. Moreover, words and phrases of the text are often too specific in representing text content preventing generic searches of information in texts. There is the difficulty of capturing the underlying concepts.

When extracting words and phrases from texts, a total lack of vocabulary control is rare, because different morphological variants of one term or different synonyms of one term are often replaced by one standard form (Lancaster & Warner, 1993, p. 84.). For instance, the least particular morphological variant (usually noun) of the terms selected is used.

3.2 Set of Controlled Language Index Terms

Assignment indexing is attributing terms to a document text from a source other than the document itself. The terms could be drawn from the indexer's head. More commonly, assignment indexing involves assigning terms or labels drawn from some form of controlled vocabulary (Lancaster, 1986; Salton, 1989, p. 230; Lancaster, 1991, p. 13 ff.; Meadow, 1992, p. 68 ff.; Lancaster & Warner, 1993, p. 80 ff.). The assigned terms are also called *descriptors.*

A controlled vocabulary is basically a predefined list of index terms constructed by some authority regarding the management of the document collection. The index terms of the list are single words or complete phrases. Usually, the vocabulary is more than a mere list. It will generally incorporate

some form of semantic structure. Two types of relationships between index terms are commonly identified: the hierarchical and associative relationships. The set of controlled language index terms is called a *classification system* (Beghtol, 1986).

Indexing with controlled language index terms assumes a predefined long-term set of users' interests (Belkin & Croft, 1992). Usually, the classification system provides a valid, often structured vocabulary for the subject content of a document collection. But, for a given document base many classification systems can be employed, possibly reflecting other aspects of the content than topicality. The classification system can vary in time and content. It always reflects a structure that for a given task hopefully over a long time is useful.

Common examples of classification systems are subject thesauri, broad subject headings, and classification schemes (Harter, 1986, p. 40 ff.). A *thesaurus* contains a variety of concepts, their equivalents, and related terms. It contains the various surface forms of concepts in texts. Thesauri are usually derived from existing and growing collections of documents in one subject discipline. The vocabulary in a thesaurus is meant to address problems of synonymy and semantic ambiguity in these collections. *Subject headings* represent the structure of the topics of heterogeneous document collections. Another type of artificial language for document representation is the very broad *classification scheme.* An example hereof, is the Dewey Decimal Classification (DDC) used in the U.S. to classify books, which is an a priori representation of all human knowledge in a great hierarchy.

Indexing with controlled language index terms has advantages and disadvantages (Harter, 1986, p. 51 ff.; Lancaster, 1986, p. 161 ff.; Svenonius, 1986).

The *advantages* especially regard the generality, the property of being unambiguous, and the preciseness of the terms. Regarding generality and the property of being unambiguous, the controlled language index terms control the variation of surface features for identical or similar concepts and thus deal with synonymy and other term relations and with semantic ambiguity (Blair & Maron, 1985; Furnas et al., 1987; Krovetz & Croft, 1992; Riloff & Lehnert, 1994). Because they are unambiguous in meaning, they are readily translated in other languages for use in applications that retrieve texts across languages. Moreover, because the terms represent general access points to text classes, they are easily employed in generic searches (Harter, 1986, p. 41 ff.), in document routing and filtering according to general classes (Belkin & Croft, 1992), in linking texts (Agosti, 1996), or in constructing topic maps of texts (Zizi, 1996). An initial classification of texts often precedes an information extraction task, so that the correct set of class-

specific natural language processing techniques can be used (DeJong, 1982; Young & Hayes, 1985; Liddy & Paik, 1993). Regarding preciseness, the controlled language phrases often function as *precoordinated index terms* that indicate and standardize specific relations between the content words of the phrases (Salton & McGill, 1983, p. 58; Soergel, 1994). For instance, a text can be indexed with the precise phrase "solvents, effects on color spectra of dyes". Controlled language index terms are useful when the texts can be represented by accurate and unambiguous concepts, irrespective of their being general or specific.

Controlled language index terms also have *disadvantages.* They permit only a few access points to a text or to represent a few perspectives. Moreover, they are rather inflexible to adapt to the needs of the users of the texts. So, the vocabulary must be regularly updated to account for changes in interest and search concepts, and changing document collections. When the vocabularies are not interchangeable, retrieval systems based on controlled language index terms are less portable and less compatible across different collections. Controlled language index terms can complement the natural language index terms in a text representation (Hearst, 1994).

Relation with text classification and categorization

Indexing with a controlled language vocabulary is related to text classification (Lancaster, 1991, p. 14 ff.). The term “classification” refers to the process of grouping entities. *Text classification* refers to the formation of text classes that are conceptually closely related. The classes often contain texts that treat the same subject. The term “*text categorization*” is used for the classification of textual documents with respect to a set of one or more pre-existing category labels or controlled language index terms by which the classes are identified.

Class assignment is *binary* (a text is or is not a member of the class) or *graded* (a text has a degree of class membership) (Sparck Jones, 1973; Cleveland & Cleveland, 1990, p. 112). The latter corresponds to the assignment of weights to the controlled language index terms.

Exactly *one descriptor* or *multiple terms* are assigned to one text. But, the number of terms assigned is usually restricted (Salton, 1975b, p. 17). When multiple terms are used, a text can belong to different related and unrelated classes. The index terms that are keys to these classes are dependently or independently assigned. The former is the case when, for instance, index terms of a hierarchical classification system are assigned: The assignment of one term involves the assignment of terms that are higher in the hierarchy. The classes – even the ones of a same level in a hierarchical classification

system – are usually not mutually exclusive (Harter, 1986, p. 56). The division of the real world into genus, species, subspecies, etc. does not always result in distinct classes. Often a better indexing is obtained by independently assigning each index term, especially the ones of the same level of a hierarchy.

3.3 Abstract

Another important form of a text representation is the *abstract* or *summary*. A summary is a condensed derivative of the source text. A summary is concerned about content information and its expression. There are many different forms of summaries (Sparck Jones, 1993; Rowley, 1988, p. 11 ff.). Usually, abstract and summary are considered as synonyms and will be used likewise throughout this book. However, sometimes a slight distinction is made. Then, an abstract rather refers to a stand-alone document surrogate (e.g., abstracts in technical journal literature), while the summary is an inherent part of the document text, which stresses its salient findings.

A minimal function that a summary should provide is being indicative of the text's content. An *indicative abstract* helps a reader to decide whether consulting the complete document will be worthwhile. An *informative abstract* reports on the actual content of the text and presents as much as possible the information contained in it. Such an abstract can act as a stand-alone text surrogate. An *extract* is composed of pieces of text extracted from the original and may have an indicative as well as an informative function. Sections or fragments of the text represent its content and/or its flavor, or highlight significant information. The latter type of abstract is called a *highlight abstract*. An abstract consisting of keywords serves as a crude indicator of the subject scope. The content of a text can be summarized in a *profile*. A profile is a frame-like representation containing distinct slots that each has a well-defined semantic meaning. The slots are filled with information from the text. A *critical abstract* not only describes a text's content, but also evaluates its content and its presentation. A *comparative abstract* evaluates a text's content and presentation with those of other texts, or represents the summary of multiple document texts.

The information content of an abstract is usually expressed in coherent text. As seen above, some abstract types present the information in other forms ranging from an extract and profile to a list of index terms.

An abstract is *highly valued* as a condensed and comprehensible representation of a text's content. It is especially appreciated by human readers for assessing the relevancy of the original text (Rowley, 1988, p. 12 ff.).

Relation with text indexing

Text indexing and abstracting are closely *related* (Lancaster, 1991, p. 5 ff.; Sparck Jones, 1993; Sparck Jones & Galliers, 1996, p. 28). The abstractor writes a narrative description of the content of a document text, while the indexer describes its content by using one or several index terms. But, the many forms of abstracts make this distinction more and more blurred. A brief summary may serve as a complex structured index description, which provides access to the text collection, while a list of key terms may serve as a simple form of abstract. Many forms of text representations are intermediate forms of indexing descriptions and abstracts, Abstracts are supposed to be more exhaustive in representing content than an indexing description (Cleveland & Cleveland, 1990, p. 105; Stadnyk & Kass, 1992).

4. INTELLECTUAL INDEXING AND ABSTRACTING

4.1 General

Historically, and still to a large extent today, text indexing or abstracting is done manually – one should say intellectually – by experts. Automatic indexing and abstracting could learn from the human cognitive processes. This does not mean that a complete cognitive process must be duplicated in automated systems, but it might be that good engineering solutions to some indexing and abstracting problems lie within some work done in the cognitive domain. First, *cognitive psychology* can offer basic contributions to textual content analysis, especially in understanding the complex mechanism of knowledge acquisition and structuring. Also, the *instruction manuals* available to indexers and abstracters might be helpful. This is the reason to briefly describe intellectual indexing and abstracting in the book.

Intellectual indexing and abstracting are not simple processes. This is the reason why trained and experienced specialists, i.e., professional indexers and abstracters, perform these tasks (Lancaster, 1991, p. 104). In some cases, the author of a text can be responsible for these tasks. But, an author often is not sufficiently trained to objectively and correctly index or abstract his or her texts (Rowley, 1988, p. 23).

Indexing or abstracting involves three major steps (Lancaster, 1991, p. 8 ff.) (Figure 1). First, there is the conceptual analysis of a source text and the identification of its content (*content analysis*). Indexing as well as abstracting is always reducing the content to its essentials and often involves *selection and generalization of information*, which form the second step of the process. Thirdly, there is the *translation* of the selected and generalized content into the language of the text representation, i.e., a particular vocabulary of index terms or a summary text. Content identification and selection of information are not always distinct steps.

4.2 Intellectual Indexing

There are many guidelines for intellectual indexing (Borko & Bernier, 1978; Rowley, 1988; Cleveland & Cleveland, 1990; Lancaster, 1991).

Content analysis

When indexing with terms extracted from or assigned to a text, the indexer usually does not perform a complete reading of the document text. A combination of *reading and skimming* is advocated. The parts to be carefully read are those likely to tell the most about the contents in the shortest period of time (e.g., summary, conclusions, abstract, opening paragraphs of sections, opening and closing sentences of paragraphs, illustrations, diagrams, tables and their captions). These salient sections are often cued by the schematic structure of the text. The rest of the text is usually skimmed to ensure that the more condensed parts give an accurate picture of what the text is about.

An important aspect of content identification is identifying the subjects of the text. Indexers have guidelines for the analysis of the *subject content* (the topics or aboutness) (Hutchins, 1985). Indexers must especially be aware of the linguistic cues that signal the thematic structure of a text on a micro as well as on a macro level (cf. chapter 2). On a macro level the notion of topic appears to be related to a text paragraph that has most links to other paragraphs. Or, a topic often appears in the first sentence of a paragraph. On a micro level, it is suggested that the theme-rheme articulations of sentences provide clues to the global topics of a text. A topic is also signaled by a noun phrase that numerous times appears as the subject of a sentence. It is also suggested that indexers first scan texts for particular words or phrases (e.g., “were killed” in the domain of terrorism) (Hutchins, 1985; Riloff & Lehnert, 1994). Then as a second step, the reader needs sometimes to evaluate the

context of the expression in case of semantic ambiguity (e.g., the context "soldiers were killed", is not anymore consistent with the terrorist domain, since victims of terrorists must be civilian).

Selection and generalization

Once the topics of the text are identified, specific topics or information can be selected. The topics can be replaced by more general concepts.

Translation of content into index terms

In a next step the identified content of the text is translated in a set of index terms. These index terms are natural language terms extracted from the text or controlled language terms selected from a classification scheme.

Indexers identify *natural language terms* in the document text, when they feel that they accurately reflect the identified content. Presumably, they are influenced by the frequency by which a content word or phrase appears in the text, by the location of its appearance (e.g., in title, in summary, in captions to illustrations) and its context (Lancaster, 1991, p. 221). Usually, indexers feel good with such a practice, which is carried out rapidly decreasing the cost of indexing. But, the guidelines are often insufficiently precise to govern the indexer's choice of appropriate subject terms from the text so that even trained indexers become inconsistent in their selection of terms (Blair & Maron, 1985).

More frequently, indexers assign *controlled language index terms* to document texts. Beghtol (1986) has described this cognitive process. It first requires the design of a classification system of index terms or category labels that will be imposed upon the documents. The actual indexing process is the mapping of natural language surface expressions of the text into the appropriate classificatory notations or index terms according to the indexer's perception of the text's content. The concept expressed by the natural language expression must be sufficiently important. So, the indexer would assign an index term to a combination of words or phrases that tend to occur frequently in the document text (Lancaster, 1991, p. 225). This sounds simple, but the concepts expressed by the controlled language index terms often occur in many variant combinations of words and phrases with variant co-occurrence frequencies. For instance, if "AIDS" occurs 20 times in a journal article, the index term "AIDS" should almost certainly be assigned. Suppose on the other hand, that "AIDS" occurs only twice in the document, but "human immunodeficiency virus" occurs a few times and "viral infection" occurs rather frequently. Then, the term "AIDS" could also be

assigned. Another example illustrates the importance of co-occurrence frequencies. If the words “heat”, “lake”, and “pollution” all occur a few times in a document, this might be enough to cause the terms “thermal pollution” and “water pollution”, to be assigned. But, “heat” and “lake” without the appearance of “pollution” would have to occur together in a document many times before “thermal pollution” would be a good bet for assignment. It is interesting to note that indexers sometimes reason by appealing to the similarity of new and old instances of texts. So, when assigning controlled language index terms, they look for textual patterns that occur in texts previously classified by these labels and assign the terms when sufficient similarity between the old and the new texts is present (Hayes-Roth & Hayes-Roth, 1977).

Indexers may attribute a *weight* to the natural and controlled language index terms based upon their judgement of term importance.

4.3 Intellectual Abstracting

Because the ability to summarize information is a necessary part of text understanding and text production, the work of Kintsch and van Dijk regarding text comprehension and production is important to unravel the intellectual process of abstracting (Kintsch & van Dijk, 1978; van Dijk & Kintsch, 1983). Many models and guidelines for intellectual abstracting exist (Borko & Bernier, 1975; Hutchins, 1987; Rowley, 1988; Lancaster, 1991; Pinto Molina, 1995; Cremmins, 1996; Endres-Niggemeyer & Neugebauer, 1998). Some of them are based upon the findings of Kintsch and van Dijk.

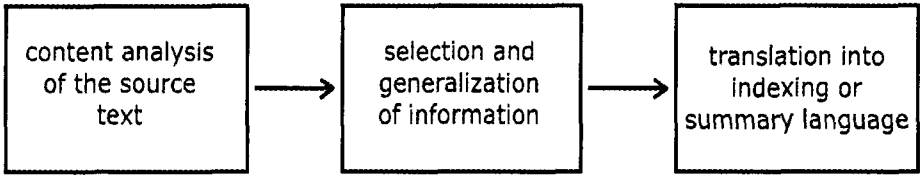


Figure 1. Intellectual indexing and abstracting.

Content analysis

Content identification for abstracting is very similar to the intellectual process of indexing. The professional abstractor learns to *skim* a text to identify the salient points quickly, followed by a more detailed *reading* of some key sections. The schematic structure of a text hints salient sections. The guidelines for making the summaries often refer to specific text types and their superstructure. A content analysis for abstracting goes into more informational detail than when indexing with terms. But, this of course is also dependent upon the type of abstract that is to be realized.

Selection and generalization

The *Kintsch and van Dijk model* of text comprehension (Kintsch & van Dijk, 1978; van Dijk & Kintsch, 1983) emphasizes the significance of the thematic structure when selecting topical information, and stresses the importance of generalizing a text's content. In this model the topics of a text are derived by applying different rules. The first regards deletion of unnecessary and irrelevant information (e.g., detailed descriptions, background information, redundant information, and common knowledge). The second bears upon selection by extracting the necessary and relevant information (e.g., information in key sections, thematic sentence selection). The selected topic segments then are stated in the form of propositions. The third rule of their model of summarization regards generalization and defines the construction of general propositions from the more specific ones. For instance, from the propositions that describe girls playing with dolls and boys playing with train sets, a description is derived of children playing with toys. A fourth rule, which is necessary in narrative texts, replaces sequences of propositions by single propositions expressing self-contained events. When summarizing the topics of a text, it is important to retain the topic emphases of the original and to make clear a distinction between the major and minor topics.

Summary production

Professional abstracting involves translating the selected and generalized content into a coherent and clear summary. This step is absent when the summary consists of phrases, sentences, or other textual units extracted from the original text.

The major concern is *brevity* and *readability* of the summary (Rowley, 1988, p. 25 ff.; Lancaster, 1991, p. 97 ff.). Usually, abstracters make a draft

that is revised and improved with the help of checklists. However, a complete reformulation of the selected information is not always desired, because of the danger of distorting the meaning of the original text (Endres-Niggemeyer, 1989). When the full-text of the abstract is used as a document surrogate in search engines, another concern is the *searchability* of the abstract. For instance, it is advised that it contains many unambiguous content terms and their synonyms (Rowley, 1988, p. 31; Lancaster & Warner, 1993, p. 88).

There are guidelines for the length of an abstract. When the abstract is a coherent text, its length is defined by different factors. The most important one is the amount of *informational detail* of the content of the source that will be provided by the abstract. A second factor is the *length of the original text.* When the abstract is a balanced picture of the most important content of the text, an ideal length is 10% to 15% of the original (Edmundson, 1964; Borko & Bernier, 1975, p. 69; Tombros & Sanderson, 1998), or 20% to 30% of the original when more informational details are needed (Brandow, Mitze, & Rau, 1995). On the other hand, when the abstract only highlights specific information, the abstract may be very brief. Sometimes, a more or less fixed length is imposed, such as a minimum and maximum upon the number of sentences (Edmundson, 1969; Paice, 1981; Brandow et al., 1995; Tombros & Sanderson, 1998), of words (Lancaster, 1991, p. 101), or of paragraphs contained in the summary (Lancaster, 1991, p. 101). Finally, the length of the abstract is determined by its *intellectual accessibility.* Some texts might be more compactly condensed then others while leaving the comprehensibility of the abstract undisturbed.

5. USE OF THE TEXT REPRESENTATIONS

Text representations have long been stored on paper (e.g., card catalogues) or on other materials as a way to efficiently and effectively ascertain the content of the original texts. Here, we are concerned with the use of text representations in systems that store and retrieve documents or information respectively in and from a database of electronic documents. In an electronic environment, there are devices that allow the text representations to be browsed, searched, and interrogated. The two major functions of the text representation (indicative and informative of the content of the original text) largely determine the type of device for their access and use. We discuss the use of indexing descriptions and abstracts in information retrieval (and filtering) systems, question-answering or information extraction systems, and browsing systems (Figure 2). There is a current

tendency to integrate these systems for an effective access to the information in document collections (Agosti & Smeaton, 1996).

5.1 Indicative and Informative Text Representations

The result of indexing or abstracting text is a representation, which function is to be indicative or informative of the text's content.

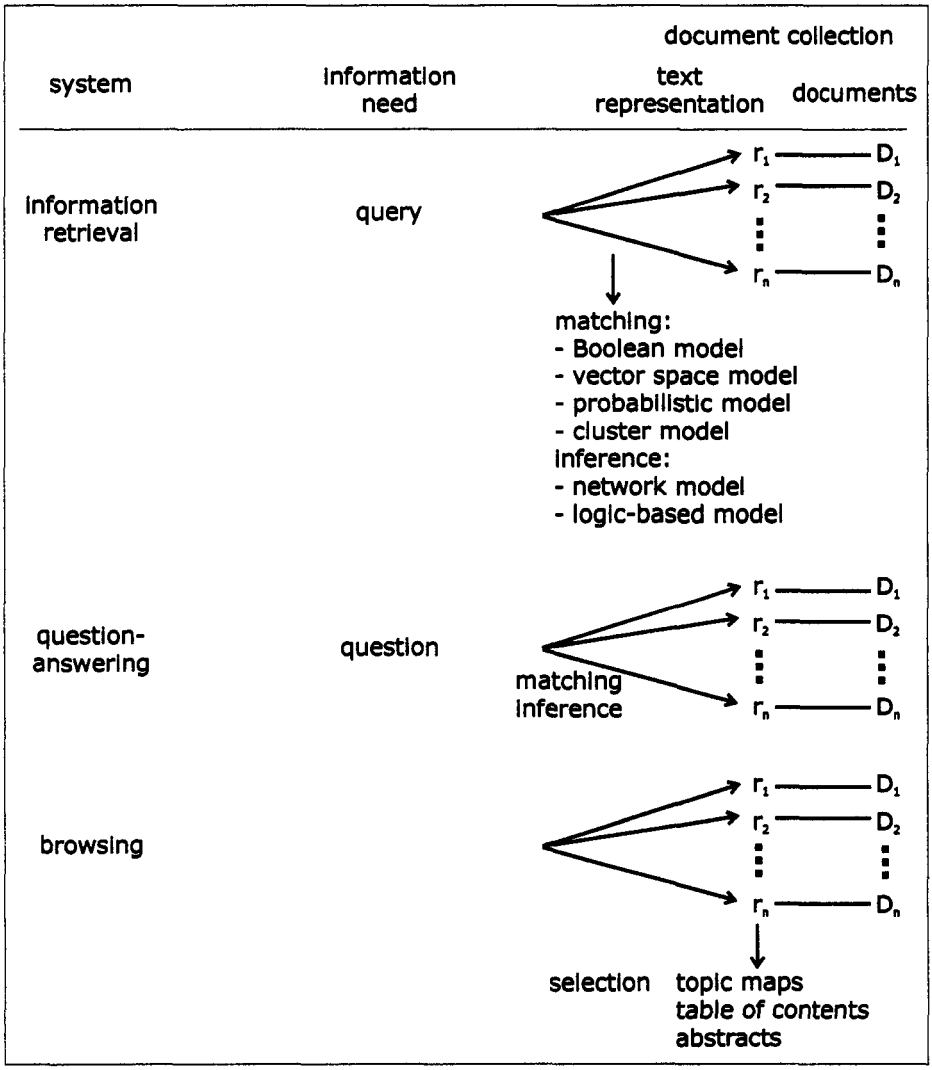


Figure 2. Actualization of an information need.

An *indicative text representation* reveals elements of the content, upon which the relevancy of the complete original text can be decided. First, it is used for *browsing* a document collection. Instead of browsing the full-texts of the documents, indexing descriptions or abstracts can be browsed and guide the user in his or her decision to see the full-text of certain documents. Second, the indicative text representation can be used in text retrieval systems. Here, its function is to filter the texts in a document collection based upon certain indicators of content. In both cases the presence and possibility of consulting the original document texts are important.

An *informative text representation* represents a surrogate of the content of the full-text or of part of the full-text. It acts, but not necessarily, as a stand-alone product without references to its original text. It is especially useful for question-answering systems. But, it is also used in information retrieval systems.

5.2 Information Retrieval Systems

A typical *information retrieval* (IR) *system* selects documents from a collection in response to a user's query, and ranks these documents according to their relevance to the query (Salton, 1989, p. 229 ff.). This is usually accomplished by matching a text representation with a representation of the query. It was Luhn (1957) who suggested this procedure.

A *search request* or *query,* which is a formal representation of a user's information need as submitted to a retrieval system, usually consists either of a single term from the indexing vocabulary or of some logically or numerically weighted combination of such terms. In case of the search request is originally formulated in natural language, a formal representation can be derived by applying simple indexing techniques or by analyzing the request with natural language processing techniques.

The abstract representations of both document text and query make an effective comparison possible. The texts, the representations of which best *match* the request representation, are retrieved. Commonly, a list of possible relevant texts is returned. In information retrieval, a rather static document collection is queried by a large variety of volatile queries. A variant form of information retrieval is *routing* or *filtering* (Belkin & Croft, 1992). Here, the information needs are long-lived, with queries applied to a collection that rapidly changes over time. Filtering is usually based on descriptions of information preferences of an individual or a group of users, which are called "*users' profiles*".

Retrieval is based on representations of textual content and information needs, and their matching. There are a number of *retrieval models* that are defined by the form used in representing document text and request and by the matching procedure. Both text and information need representations are uncertain and additionally do not always exact match. Querying an information retrieval system is not like querying a classical database. The matching is not deterministic. Retrieval models often incorporate this element of uncertainty. Moreover, retrieval models generally rank the retrieved documents according to their potential relevancy to the query. This is why they are sometimes called ranking models (Harman, 1992a). Because the retrieval models developed in an environment with documents that were manually indexed with a set of terms, many models rely upon this form of text descriptors. In the following, we give an overview of the most common retrieval models.

The Boolean model

In the oldest model, the *Boolean retrieval model* (Salton, 1989, p. 235 ff.; Smeaton, 1986), a query has the form of an expression containing index terms and Boolean operators (e.g., “and”, “or”, “not”) defined upon the terms. The retrieval model compares the Boolean query statement with the term sets used to identify document content. A document the index terms of which satisfy the query is returned as relevant. This retrieval model is still employed in many commercial systems. It is a powerful retrieval model, when the users of the retrieval system are trained in designing Boolean queries. In the pure Boolean model, no ranking of the documents according to relevance is provided. Variants of the model provide ranking based upon partial fulfillment of the query expression.

The vector space model

In the *vector space retrieval model* (Salton, 1989, p. 313 ff.; Wang, Wong, & Yao, 1992), documents and queries are represented as vectors in a vector space with the relevance of a document to a query computed as a distance measure. Both query and documents are represented as *term vectors* of the form:

$$D_m = (a_{m1}, a_{m2}, \ldots, a_{mn})$$
$$Q_k = (q_{k1}, q_{k2}, \ldots, q_{kn})$$

where the coefficients a_{mi} and q_{ki} represent the values of index term i in document D_m and Q_k respectively. Typically a_{mi} (or q_{ki}) is set equal to 1 when term i appears in document D_m or query Q_k respectively and to 0 when the term is absent (vectors with binary terms). Alternatively, the vector coefficients could take on numeric values indicating the weight or importance of the index terms (vector with weighted terms). As a result, a document text and query are represented in an *n-dimensional vector space* $\Re$ (with n = number of distinct terms in the index term set of the collection).

Comparing document and query vector is done by computing the similarity between them (Jones & Furnas, 1987). The most common *similarity functions* are the cosine function, which computes the cosine of the angle between two term vectors, and the inner product, which computes the scalar product between the term vectors. The result of the comparison is a ranking of the documents according to their similarity with the query.

The vector model is very popular and successful in research settings and commercial systems because of the simplicity of the representation, its application in unrestricted subject domains and upon different text types, and simple comparison operations. It has been criticized because it does not accurately represent queries and documents (Raghaven & Wong, 1986). It adapts a simplifying assumption that terms are not correlated and term vectors are pair-wise orthogonal. However, many useful and interesting retrieval results have been obtained despite the simplifying assumptions.

The probabilistic model

The *probabilistic retrieval model* (Fuhr, 1992) views retrieval as a problem of estimating the probability that a document representation matches or satisfies a query. The term "probabilistic retrieval model" is generally used to refer to retrieval models that produce the probability that a document is relevant for the query and rank documents according to these probabilities ("Probability Ranking Principle") (Robertson, 1977; Croft & Turtle, 1992). In this view, many retrieval models can be seen as probabilistic. Often, the term specifically refers to retrieval models that learn the weight of query terms from the documents that are judged relevant or non-relevant for the query and that contain or do not contain the terms. The earliest probabilistic models that learn the weight or probability of a query term from a training corpus are described by Maron and Kuhns (1960) and Robertson and Sparck Jones (1976). The current models use more refined statistical techniques, such as 2-Poisson distributions (Robertson & Walker, 1994) and logistic regression (Gey, 1994) for estimating this probability.

When estimating the probability of the relevance of a document to a query, term independence is assumed.

Probabilistic models are in use in some commercial systems and are being actively researched.

The next two models infer the relevancy of a document from the query. The inference relies upon knowledge that reflects the properties of the subject domain, upon linguistic knowledge, and/or knowledge of the supposed retrieval strategies of a user. The knowledge contributes in building semantically rich representations of the content of document and query, It is assumed that these semantic representations help in identifying meaningful documents for the user. The inference strategy in both models is different. In the network model inference is based upon the combination of evidence as it is propagated in a network. In the logic-based model logical rules are used to deduce the relevancy of a document for a query. Both models provide the possibility of reasoning with uncertainty. Their major bottleneck is acquiring and implementing the knowledge bases.

The network model

In the *network retrieval model* (Croft & Turtle, 1992; Turtle & Croft, 1992) document and query content are represented as networks. Estimating the relevance of a document is accomplished by linking the query and document networks, and by inferring the relevancy of the document for the query. The model is also well suited to reason with uncertain information: *Bayesian networks* are used for probabilistic representation of the content of documents and query and for probabilistic inference (Del Favero & Fung, 1994; Fung & Del Favero, 1995).

Networks are very well suited to represent structure and content of documents and queries. The networks have the form of *directed acyclic graphs* (type DAG). The inference network model is popular in information retrieval. In typical cases, the nodes of the document network represent identifiers, concepts, or index terms. Each document typically has a text node, which corresponds with a specific text representation and which is composed of the components that make up the representation. A document can have multiple text nodes that are generated with different indexing techniques. Intermediate levels in the representation are possible (e.g., concepts and their referring index terms in the texts). The relationships between nodes in a network may be probabilistic or weighted. Each set of arcs into a node represents a (probabilistic) dependence between the node and its parents (the nodes at the other ends of the incoming arcs). Often a document network is once built for the complete document collection. A

similar representation is generated for the query. The two networks are connected by their common concepts and form the inference or causal network.

The retrieval is a process of *inference* on the network. Especially the *Bayesian inference* applied upon multiple sources of uncertain evidence is attractive in an information retrieval context. Retrieval is then a process of combining uncertain evidences from the network and inferring a belief that a document is relevant. This belief is computed as the propagation of the probabilities from a document node to the query node. Documents are ranked according to this belief of relevance.

The logic-based model

The *logic-based retrieval model* (van Rijsbergen, 1986; Chiaramella & Chevallet, 1992; Lalmas, 1998) assumes that queries and documents can be represented by logical formulas. The retrieval then is inferring the relevance of a document for a query. The above Boolean model is logic-based. But, the typical logic-based model will use the information in query and document in combination with domain knowledge, linguistic knowledge, and knowledge of users' interests and strategies from a coded knowledge base. The knowledge will be used by the matching function as part of proving that the document implies the query.

The relevance of a document to a query is deduced by applying *inference rules.* In the logical model relevance of a document for a query is defined as: Given a query Q and a document D, D is relevant to Q if D logically implies Q (D->Q). Boolean logic is too restricted for this task. It cannot deal with temporal and spatial relationships, and especially not with contradictory information or uncertain information. In order to cope with uncertainty, a logic for *probabilistic inference* is introduced with the notion of uncertain implication: D logically implies Q with certainty P (P (D->Q)). The evaluation of the uncertainty function P is related to the amount of semantic information which is needed to prove that D->Q. Ranking according to relevance then depends upon the number of transformations necessary to obtain the matching and the credibility of the transformations. To represent uncertain implications and reason with them, modal logic is sometimes used (Nie, 1989; van Rijsbergen, 1989; Chiaramella & Nie, 1990; Nie, 1992). For instance, when a matching between query and text representation is not successful, the text representation is transformed in order to satisfy other possible interpretations (cf. the possible worlds of the modal logic) that might match the query.

In a multi-media environment, logic-based retrieval has the advantage to easily integrate text representations with other forms of document representations (e.g., logical structure, content of images) (cf. Bruza & van der Weide, 1992; Chiaramella & Kheirbek, 1996; Fuhr, Gövert, & Rolleke, 1998).

The cluster model

In the *cluster retrieval model* a query is ranked against a group of documents (van Rijsbergen, 1979, p. 45 ff.; Griffiths, Luckhurst, & Willett, 1986; Salton, 1989, p. 341 ff.; Hearst & Pedersen, 1996). The general assumption is that mutually similar documents will tend to be relevant to the same queries, and, hence, that automatic determination of groups of such documents increases the efficiency of the search of relevant documents and can improve the recall of the retrieval. Similar documents are grouped in a cluster. For each cluster, a representation is made (e.g., the average vector (centroid) of the cluster) against which a query is matched. Upon matching, the query retrieves all the documents of the cluster. Typically a fixed text corpus is clustered either to an exhaustive partition, disjoint or otherwise, or into a hierarchical tree structure. In case of a partition, queries are matched against clusters and the contents of the best scoring clusters are returned as a result, possibly sorted by score. In the case of a hierarchy, queries are processed downward, always taken the highest scoring branch, until some stopping condition is achieved. The subtree at that point is then returned as a result. Hybrid strategies are also available. Documents and query are commonly represented as term vectors. The similarity between pairs of vectors is computed with similarity functions (see above). Different algorithms for clustering the term vector of documents are available (for an overview, see Willett, 1988).

5.3 Question-Answering Systems

A *question-answering* or information-extraction *system* is a system that synthesizes an answer from one or multiple document texts. In contrast to a retrieval system that retrieves documents in which the answer can be found, a query in a question-answering system retrieves specific information from the documents. The answers are typically extracted or inferred from text representations. Question-answering systems use text representations that are real substitutes of the content of the source text or of part of that content.

The representations exhibit an explicit and regular form. They often have the form of instantiated frames, the slots of which contain the information

that is queried (e.g., Young & Hayes, 1985). Relationships can be defined between the frames. The frames form the information or knowledge base for answering the queries. When this information base is a pure collection of facts, its interrogation is like *querying a classical database*. When complex frame relationships are defined, the set of text representations constitutes a real knowledge base. The querying then is more like *inferring answers from a knowledge base*. When the relations between the frames are uncertain, a form of Bayesian inference or a logic that deals with uncertainty is necessary.

5.4 Browsing Systems

Browsing or navigation *systems* are usually part of hypertext and hypermedia systems and allow users to skim document collections in the search for valuable information. Text representations, especially text abstracts, can be part of a hypertext system. The abstracts are browsed in a sequential (as leafing through a book) or in a non-sequential way. The browsing or navigation can take place in a collection of stand-alone text representations or in a document collection with defined links between the text representations and their original texts. The advantage of browsing systems is that users need not to generate descriptions of what they want or specify in advance the topics in which they are interested, but can just indicate documents they find relevant. This way of information access is valuable when a user has no clear need, cannot express his need accurately, or is a casual user of the information (Allen, 1990; Croft, Krovetz, & Turtle, 1990; Hearst, 1994). Browsing text abstracts acts as an extra filter to documents that are retrieved as the answer of an information need and makes their selection more straightforward, accurate and faster (Tombros & Sanderson, 1998).

There are a few other ways in which text representations can be helpful parts of browsing systems. Content descriptors of text are useful components of devices that guide the user of the collection in his or her choice of documents. Their presence in topic maps or tables of contents is valuable (Cutting, Karger, Pedersen, & Tukey, 1992). In advanced systems, the text representations can help the automatic creation of links between texts that treat identical and similar contents, when linking texts that have similar representations (Lucarella & Zanzi, 1996; Salton, Allan, Buckley & Singhal, 1996).

6. A NOTE ABOUT THE STORAGE OF TEXT REPRESENTATIONS

We give a short overview of important data structures that are used for storing the text representations in order for being them to be searchable, browsable and questionable. For more detailed overviews of existing data structures, we refer to Frakes and Baeza-Yates, 1992, p. 28 ff, and to Kowalski, 1997, p. 65 ff. Two aspects of a data structure are important: the ability to represent concepts and relationships, and the ability to support the location of these concepts in the document collection.

In a retrieval application the most common form is storing index terms and their coupling to documents in an *inverted index* or an *inverted file*. For each term, the inverted file stores the identifiers or addresses of all documents that are indexed by that term. The complete inverted file is first represented as an array of indexed documents, where each row represents an address of the document and each column the assignment of a particular term to the document (binary value indicating the presence or absence of a term, or term weight). The document-term array is then transposed (so-called *inverted*) in such a way that each row of the transposed array specifies the documents corresponding to some particular term. Information about the location of a term in a document can be added.

Another searchable data structure in a retrieval environment is the *n-gram structure* that breaks the words and phrases of the text representation into smaller string units of *n* characters and uses these fragments for search. This allows searching different morphological forms of words.

Signature files contain signatures or bit patterns, which represent the index terms of documents. The signatures can be efficiently searched. In a common signature method, documents are split into logical blocks each containing a fixed number of index terms. Each word in the block is hashed to give a signature, which is a bit pattern with some of the bits set to 1. The signatures of each word in a block are OR'ed together to create a block signature. The block signatures are then concatenated to produce the document signature. Searching can be efficiently done by comparing the signatures of queries with the ones of documents.

In question-answering systems, the text representations can be stored as a set of facts in a database. More often, they are stored as frames in a knowledge base and used by knowledge-based systems or expert systems. Because of portability and ease of maintenance of the knowledge, knowledge bases are usually stored separately from the inference mechanism that reasons with the knowledge.

In a browsing environment, a text representation is often stored as hypertext in the HyperText Markup Language (HTML) and linked to the original text. HTML defines the internal structure for information exchange across the World Wide Net on the Internet. It defines a markup language for layout and display of the hypertext and for defining links between textual objects. The hypertext reference can be an anchor indicating the text position, when the referenced original text is stored on the same file as its representation. It can also be a file name, when the referenced item is stored on the same machine as the referencing representation, or an URL (Uniform Resource Locator), which specifies an access protocol, the Internet address of the server where the item is stored, and the file name of the item.

7. CHARACTERISTICS OF GOOD TEXT REPRESENTATIONS

The ultimate *aim of indexing* and *abstracting* is to increase recall, the proportion of relevant documents that are viewed or retrieved in a browsing and retrieval system respectively, and to increase precision, the proportion of viewed or retrieved documents that are relevant (cf. Salton, 1989, p. 277 ff.). A high recall in a question-answering system refers to a high proportion of correct answers given the available answers, while a high precision concerns a high proportion of correct answers among the answers (Chinchor, 1992; Chinchor, Hirschman, & Lewis, 1993). A text representation that is the result of indexing and abstracting has a number of characteristics in order to increase recall and precision of selected documents or information. Depending upon the application, each characteristic has a varying degree of importance. Some of these characteristics can be described solely by referring to the original source text. Others are defined in relation to the other text representations in the document collection. The following outlines some important characteristics, some of which represent conflicting demands.

1. A major characteristic of the text representation is the ability to represent the *aboutness* or the topics of a document text (Maron, 1977; Hutchins, 1985). Topic identification is highly valued in browsing, retrieval, and filtering systems, especially when these systems operate in general settings (e.g., public libraries, Internet). Besides aboutness is the ability to represent the *potential meanings* that a text has for its users (Hutchins, 1977; Salton & McGill, 1983, p. 54; Hutchins, 1985; Lancaster, 1991, p. 8; Fidel, 1994). This might be realized by a more detailed indexing or

abstracting resulting in a representation of the subtopics and of specific information of the source text. This "user orientation" in indexing and abstracting allows a fine-grained selection of topical content. This property is highly valued in information retrieval systems that are used by specialists and experts (e.g., research libraries, databases of medical documents) and in question-answering systems.

2. In contrast to the foregoing, a text representation is often a *reduction of the content* of the original text. This reduction can be the result of a generalization or of a selection of the content. This characteristic is important when retrieving or filtering information from large document collections (Sparck Jones, 1991). When indexing descriptions or abstracts are used as text previews in browsing or navigation systems, this reductive character is also fundamental.
3. It is not enough for a text representation to be a good description of the content of the source text. It should allow differentiating its content from the contents of other text representations (Lewis & Sparck Jones, 1996). This characteristic is especially useful in browsing and retrieval systems, when the text representation has to *discriminate* relevant documents from the many non-relevant ones. If the text representation reduces the content, it naturally reduces the difference with other text representations. Again, being discriminative and being reductive do not always go hand in hand.
4. When browsing large document collections or retrieving information from them, it is important to consult all relevant documents. In these collections, when similar text representations are grouped, texts can be efficiently retrieved or consulted with a high degree of recall (cf. the clustering retrieval model) (Lewis & Sparck Jones, 1996). In this case, text representations must contain content elements that allow *grouping*. This characteristic also conflicts with the foregoing requirement of being discriminative.
5. Finally, a text representation *normalizes* lexical and conceptual variations of the source text (Hutchins, 1975, p. 37 ff.). This characteristic is advantageous in information retrieval and filtering systems, and especially important in question-answering systems.

Text representations themselves are judged by the criteria of exhaustivity, specificity, correctness, and consistency (Salton & McGill, 1983, p. 55; Lancaster &Warner, 1993, p. 81 ff.; Soergel, 1994).

1. *Exhaustivity* refers to the degree to which all the concepts and notions included in the text are recognized in its description, including the central topics and the ones treated only briefly.
2. *Specificity* refers to the degree of generalization of the representation.
3. *Correctness* is important. Indexing and abstracting are susceptible to two kinds of errors: errors of omission and errors of commission. The former refers to a content description that should be assigned, but is omitted. The latter refers to a content description that should not be assigned, but is nevertheless attributed. Omitting a correct description and assigning a broader, narrower, or related description is a special kind of error that is at once an error of omission and commission. Correctness compares the actual text representation with the ideal one.
4. *Consistency* compares representations that are made of the same source text in different contexts (e.g., generated by different techniques).

When *evaluating automatic indexing and abstracting*, exhaustivity, and specificity are difficult to quantify. Current evaluation emphasizes correctness and consistency. Automatic text indexing and summarization are usually seen as natural language processing tasks. The criteria applied in performance evaluation of such tasks normally fall under two major heads, intrinsic and extrinsic (Sparck Jones & Galliers, 1996, p. 19ff.). *Intrinsic criteria* are those relating to a system's objective, *extrinsic criteria* are those bearing upon its function, i.e., to its role in relation to its setup's purpose. It often depends upon the type of text representation whether the evaluation is intrinsic or extrinsic. For instance, the value of extracted natural language index terms, is usually measured by computing the recall and precision of the retrieval of texts based on representations that contain the terms, which is an extrinsic evaluation. On the other hand, controlled language subject and classification codes are judged by measuring the recall and precision of the assigned terms as compared to their manual assignment by experts, which is an intrinsic evaluation. When discussing the methods of automatic indexing and abstracting in the next part, evaluation will be shortly described with each major approach. It is agreed upon that evaluation of text indexing and abstracting needs further research (cf. Hersh & Molnar, 1995).

The idea of an exhaustive, multi-functional text representation for managing document texts is appealing. It allows producing multiple *views* of the same text and consequently selecting specific information conforming to different needs (cf. Soergel, 1994; Lucarella & Zanzi, 1996; Frants, Shapiro, & Voiskunskii, 1997, p. 139 ff.). Additionally, when the content attributes have weighted values that reflect content importance, it allows zooming in and out into informational detail of a text's content (cf. Fidel, 1994). At

different levels of informational detail, one might discriminate text representations from others in the collection, or, if needed, group representations, Such an exhaustive text representation can combine different types of content representations (e.g., natural language and controlled language index terms, extracted words, phrases, and other informational units) (cf. Strzalkowski et al., 1997). New forms of text representations will certainly be tested in the future.

8. CONCLUSIONS

In this chapter we described the traditional forms of indexing descriptions and abstracts and their advantages and disadvantages. We also outlined the intellectual process of indexing and abstracting and saw that these cognitive processes heavily rely upon text structures and reoccurring word patterns to identify the content of texts. In chapter 1 we extensively elaborated the need and the increasing importance for systems that automatically produce useful and correct text representations in the form of indexing descriptions and abstracts. The use of the text representations in browsing, retrieval, and question-answering systems confirms this necessity.

In the next part we give a detailed overview of existing techniques for automatic indexing and abstracting. They comprise techniques for identification of key terms (natural language index terms) in texts, assignment of fixed descriptors (controlled language index terms) to texts, and methods for text summarization.

This page intentionally left blank.

PART II

METHODS OF AUTOMATIC INDEXING AND ABSTRACTING

This page intentionally left blank.

Chapter 4

AUTOMATIC INDEXING: THE SELECTION OF NATURAL LANGUAGE INDEX TERMS

1. INTRODUCTION

The majority of existing automatic indexing methods select natural language index terms from the document text. The index terms selected concern single words and multi-word phrases and are assumed to reflect the content of the text. They can be directly extracted from the title, abstract, and full-text of a document. It was Luhn (1957) who first suggested that certain words could be automatically extracted from texts to represent their content. Still today, the search engines that operate on the Internet index the documents based upon this principle (Szuprowicz, 1997, p. 43 ff.). However, not all words in a text are good index terms and words that are good index terms do not contribute equally in defining the content of a text. A number of techniques help in identifying and weighting reliable content terms.

A prevalent process of selecting natural language index terms from texts that reflect its content is composed of the following steps (cf. Salton, 1989, p. 303 ff.):

1. the identification of the individual words of the text, called lexical analysis;
2. the removal of function words and highly frequent terms in the subject domain that are insufficiently specific to represent content using a stoplist;
3. the optional reduction of the remaining words to their stem form, called stemming;

4. the optional formation of phrases as index terms;
5. the optional replacement of words, word stems, or phrases by their thesaurus class terms;
6. the computation of the weight of each remaining word stem or word, thesaurus class term, or phrase term.

A variant ordering of the above steps is possible. For instance, recognition of phrases can occur before removal of function words. Before discussing the different steps, evaluation of the selected natural language index terms is shortly described. We finish this chapter by enumerating the accomplishments and problems of the techniques. Because the replacement of words or word stems by their thesaurus class terms concerns indexing with a controlled language vocabulary, we discuss this item in the next chapter.

2. A NOTE ABOUT EVALUATION

The selection of natural language index terms is commonly evaluated in an extrinsic way (cf. Sparck Jones & Galliers, 1996, p. 19 ff.). Extrinsic evaluation judges the quality of the index terms based on how the index terms perform in some other task. It is usually measured how they affect *retrieval effectiveness* when document selection is based upon these terms. Retrieval effectiveness is usually measured in terms of *recall* and *precision*:

$$\text{recall} = \frac{\text{number of relevant documents retrieved}}{\text{total number of relevant documents in the collection}}$$

$$\text{precision} = \frac{\text{number of relevant documents retrieved}}{\text{total number of documents retrieved from the collection}}$$

3. LEXICAL ANALYSIS

Lexical analysis starts when the text is already electronically stored and can be regarded as a sequence of characters. Lexical analysis is the process of converting an input stream of characters into a stream of words or tokens (Fox, 1992). A word or token is defined as a string of characters separated by white space and/or punctuation. Lexical analysis produces candidate index terms that can be further processed, and eventually selected as index terms.

The *recognition* of individual words seems a simple process, but it is not always straightforward.

1. *Abbreviations* (e.g., "cf.") may be confused with words ending with a full stop at the end of a sentence.
2. A difficult and language dependent decision is whether to break *hyphenated terms* into their constituent words or to keep them as single index terms. In English as well as in Dutch, some compound words can be formed with hyphenating. Separating the hyphenating terms increases recall, when the terms are used for retrieval, but decreases precision. Also, hyphens may be part of a proper name (e.g., "MS-DOS") or may split a single word into syllables at the end of a line in a hyphenated text.
3. *Numbers* in texts usually do not make up good index terms and are often neglected.

During lexical analysis, it is common to make *small transformations* to the words.

1. The *case of letters* is usually not significant in index terms and all characters can be converted to either lower or upper case. Caution must be taken with proper name phrases. Preserving case distinctions of index terms usually enhances precision, but decreases recall of a search.
2. *Abbreviations* and *acronyms* may be transformed to their original format using a machine-readable dictionary.

Lexical analysis is extensively studied for text processing tasks. A widespread approach is treating the lexical analyzer as a *finite state automaton* or *finite state machine* (Aho, Sethi, & Ullman, 1986, p. 113 ff.). A *finite state machine* (Krulee, 1991, p. 167 ff,) is the most simple machine to recognize if a certain input string is allowed in the pre-defined syntax of a language. When parsing the input string, the finite state machine starts from an initial state or configuration, then by means of moves, which define the transition from one state to another, the machine reads sequences of the input until a final state is attained and the input string is completely processed. In a finite state machine only a finite number of states are defined. Finite state machine based lexical analyzers recognize individual words by reading the stream of input characters until some character other than a letter or digit is found. When reading a character or word, it may be changed (e.g., replaced by another character or word). Such translation information can be encoded in tables, or in flow of control.

4. USE OF A STOPLIST

The words of a text do not have equal value for indexing purposes. A *stoplist* or *negative dictionary* is a machine-readable list of words (*stopwords*) that can not be chosen as index terms (Salton, 1975a, p. 30 ff.; Salton, 1989, p. 279; Fox, 1992). Commonly, a stoplist is used to eliminate words that do not bear upon the content of the text. When done early in the indexing process, elimination of stopwords has the extra advantage of making further processing of the candidate index terms more efficient and reducing their storage space. Stoplists vary in size (e.g., most stoplists in English contain from about 50 to 400 words).

There are different techniques for building a stoplist.

1. Some *word classes* are better indicators of the content of a text, while others contain function words that serve grammatical purposes and do not refer to objects or concepts (e.g., "the", "and", "of') (see chapter 2). Function words make up a large fraction of texts. It is critical to eliminate them as index terms. Words belonging to the syntactic classes that comprise function words form a *generic stoplist* (e.g., Hoch, 1994). An inverse strategy selects words as index terms when they belong to a specific syntactic class (e.g., nouns) (Luhn, 1957; Prikhod'ko & Skorokhod'ko, 1982).
2. The most prevalent way to construct a stoplist is to include the words that most frequently occur (Luhn, 1957; Salton, 1989, p. 279). This is based upon the finding that the frequency of occurrence of a function word is much higher than the frequency of occurrence of a content word. Either the stoplist is constructed by considering the most frequent words of a general corpus that reflects a broad range of subjects resulting in a *generic stoplist* (e.g., a stoplist for the English language obtained from the Brown corpus: Fox, 1989), or it is constructed by observing the frequency of words of the document collection that is to be indexed resulting in a *domain-specific stoplist.* A threshold value is set to determine the number of words to be included in the stoplist (e.g., 200 most frequently occurring words) or to define their minimum frequency of occurrence. In rare cases, words with a very low *inverse document frequency weight* (see below) are considered as stopwords. However, frequency of occurrence of a word in a document corpus is not a 100% sound criterion for content importance. For instance, it is possible that words that frequently occur in a corpus are important index terms. This is the case when a subset of the document database contains multiple texts

treating the same topic. It is also possible that a specialized text database contain words useless as index terms that are not frequent in standard language nor in the database.

3. Because function words tend to be small, occasionally all *short words* that contain less than a threshold value number of characters are removed from the text (Ballerini et al., 1997). Using an anti-stopword-list then prevents the removal of important short words.
4. An more aggressive method for removal of domain-specific stopwords uses a collection of *training texts* and information about their *relatedness* in the training set (Wilbur & Sirotkin, 1992; Yang & Wilbur, 1996). A word score reflects how important the word is in identifying texts that are related to each other (i.e., texts that treat the same topic). This score is computed based on word distribution over related texts. Stopwords are words with a low score.

The creation of a stoplist is a process that occurs before the actual indexing of individual texts. *Removing stoplist words* during automatic indexing can be treated like a search problem. A potential index term is checked against the stoplist and eliminated as candidate index term if found there. Searching a stoplist is more efficient by binary search or hashing. Stopword removal is often integrated in the lexical analyzer (Fox, 1992).

5. STEMMING

Another technique that may improve the quality of automatic indexing is *stemming.* Stemming or conflating words is the process of reducing the morphological variants of the words to their stem or root (e.g., mapping singular and plural forms of a same word to a single stem). The program that executes the mapping is called a *stemmer.* It is assumed that words with the same stem are semantically related and have the same meaning to the user of the text.

Stemming in the field of information retrieval aims at *improving the match between the index terms of query and document text.* The chances of matching increase when the index terms are reduced to their word stems. Stemming, thus, is a recall-enhancing device to broaden an index term in a text search (Salton, 1986). Additionally, stemming reduces the number of index terms by mapping the morphological variants to a standard form. Consequently, the size of the text representation decreases, which is beneficial in terms of storage.

There are four major automatic approaches to stemming.

1. The *table lookup* method is the simplest method and requires the terms and their stems to be stored in a table or a machine-readable dictionary (Frakes, 1992). Stemming is done via lookups in the table. The advantage of this method is that the stemming results are generally correct. However, the table becomes large, when it takes into account terms in standard language and possibly terms in the specialized subject domain of the text corpus. Large tables require large storage spaces and efficient search algorithms (e.g., binary search tree, hash table).
2. *Affix removal algorithms* are most commonly used and remove suffixes and/or prefixes from terms leaving a stem (Frakes, 1992). These algorithms also transform the resultant stem (e.g., ‘a’ to ’u’ in “ran” to “run”; cf. in Dutch: "ie" to "oo" in "liep" to "loop"). The Lovins stemmer (1968) removes suffixes using a longest match algorithm. It removes the longest possible string of characters from a word according to a set of rules. This process is repeated until no more characters can be removed. Even after all characters have been removed, stems may not be correctly conflated. Then, linguistic knowledge is employed to recode the stem. The Porter algorithm (Porter, 1980) removes affixes by applying a set of rules. The rules also account for transformations of the stem. Affix removal algorithms can become quite ingenious and employ many inferences from linguistic knowledge about the internal structure of words for generating the correct reductions (Krovetz, 1993). The knowledge that the affix removal algorithms employ is language dependent.
3. *Letter successor variety stemmers* (Hafer & Weiss, 1974) learn morphemes from a large body of example words. They use the frequencies of letter sequences in a corpus of texts as the basis of stemming. For each possible begin sequence of letters of a word the number of variant successor letters (distinct letters) in the corpus is computed. The successor variety tends to decrease from left to right, while at boundaries of morphemes (e.g., after an affix) the successor variety rises. By calculating the set of successor varieties for a test word and noting the peaks, we can detect the morphemes of a word. When at the end of a word the successor variety becomes very low, suffixes are detected by considering the word and the words in the corpus in reverse letter order. Heuristics determine whether a found morpheme is a stem or an affix. When the morpheme matches other corpus words, it is probably a stem. When the segment occurs as first (last) part in a number of different words, it is probably a prefix (suffix). The advantage of this method is that it can adapt to changing text collections and languages, but the method does not distinguish inflectional from derivational affixes.

4. Finally, the *n-gram method* conflates terms based on the number of *n*-grams they share. An *n*-gram is a sequence of *n* consecutive letters. Adamson and Boreham (1974) compute the number of unique matching bigrams in pairs of words (computed with the Dice coefficient[1]). A bigram is a pair of consecutive letters. Xu and Croft (1998) use trigrams. Terms that are strongly related by the number of shared *n*-grams are clustered into groups of related words. Heuristics help in detecting the root form (see above), or special cluster algorithms might be useful for this task (e.g., cluster algorithms based on the selection of representation objects, cf. chapter 8). Again this method does not distinguish between inflectional and derivational affixes.

Many stemmers have been developed for the English language (overview see Frakes, 1992). The two most common stemmers for English are the Lovins stemmer (Lovins, 1968) and the Porter stemmer (Porter, 1980). Kraaij and Pohlmann (1996) have used the Porter algorithm to develop a stemmer for Dutch and have developed an additional inflectional and derivational stemmer using a computer readable dictionary of Dutch words. In Dutch nominal compounds are generally formed by concatenating two (or more) words to create a single orthographic word (e.g., "fiets" + "wiel" = "fietswiel" ("bicycle" + "wheel" = "bicycle wheel")). Stemmers of the Dutch language are extended with a compound analyzer (*word splitter*) (Vosse, 1994 cited in Kraaij & Pohlmann, 1996). This tool aims at splitting a compound into its components (stems) by applying word combination rules and a lexicon.

Automatic stemming can result in *overstemming* and *understemming.* The former refers to the case when too much of the term is removed, which causes unrelated terms to be conflated to the same stem. The latter refers to the removal of too little from a term, which prevents related terms from being conflated. Stemming is useful when the morphology of a language is rich (e.g., Hungarian or Hebrew) or when the text to be indexed is short (Krovetz, 1993). Removal of inflectional morphemes usually has little impact upon a word's meaning and thus can be safely done (e.g., mapping singular and plural of a same word to a single stem). Removal of derivational morphemes may change a word's meaning. Stemming has been evaluated from the viewpoint of retrieval effectiveness (overview of the studies regarding the English language, see Frakes, 1992 and Hull, 1996; regarding the Dutch language, see Kraaij & Pohlmann, 1996). It is generally agreed upon that stemming either has a positive or no effect on retrieval effectiveness. Splitting Dutch compound nouns has been proven effective to increase retrieval performance.

6. THE SELECTION OF PHRASES

It is commonly agreed that phrases (see chapter 2) carry more semantic meaning than individual words. Especially, noun and prepositional phrases are believed to be content bearing units of information and thus *good indicators of a text's content* (Earl, 1970; Salton, Buckley, & Smith, 1990; Smeaton, 1992). A phrase can be considered as a *specification of a concept.* It may denote an important concept in certain subject domains. For instance, the term "joint venture" is an important term in financial texts, while neither "joint" nor "venture" are important by themselves. Phrases improve the specificity of the indexing language. The use of phrases as index terms increases *precision* of a retrieval operation (Fagan, 1989). Additionally, phrases are *less ambiguous* in meaning than the single words they are composed of. Each word of the phrase offers the context to remove an ambiguity in the remainder of the phrase (e.g., the word "tree" removes the ambiguity of the word "bark" in "the bark of a tree"). Despite extra computational requirements for their recognition (Callan & Lewis, 1994), phrases are prime candidates of natural language index terms to be included in a text representation.

When phrases are employed as natural language index terms, two aspects need to be automated: their identification in texts and their normalization to a standard form. There are two major techniques for identifying phrases: statistical phrase recognition and syntactic phrase recognition, each generating respectively statistical phrases and syntactical phrases (Croft, Turtle, & Lewis, 1991). Phrases that refer to the same concept can be expressed in many different ways. So, normalization of phrases to a standard form is necessary. A special case of phrase recognition concerns the recognition of proper names.

6.1 Statistical Phrases

Statistical phrase recognition assumes that, when a set of words often co-occur in the texts of a document collection, the set of co-occurring words might denote a phrase. The idea of using statistical associations between words goes back at least to the early 1960s (see Salton, Buckley, & Smith, 1990 for an overview of the research; Damerau, 1993). Often, pairs of adjacent non-stopwords are considered as candidate phrases (Salton, Yang, & Yu, 1975; Buckley, Salton, & Allan, 1992), but also sets of a few words are tested. A *statistical phrase* is then defined by constraints upon the frequency of occurrence of the phrase, upon the co-occurrence of its components, and/or upon the proximity of its components in the texts

(Salton et al., 1990; Croft et al., 1991). The proximity of phrase components can be defined by their number of intervening words or by their occurrence in the same sentence, paragraph, or whole text (Salton & McGill, 1983, p. 84 ff.). When for a given candidate phrase, the values of the above parameters are within threshold values (set after experiments with the text collection), it is selected as index term.

Occurrence frequency and proximity parameters do not always yield correct and meaningful phrases. Two or more words possibly co-occur for reasons other than being part of the same phrasal concept. It is therefore not surprising that Fagan (1989) found that the use of statistical phrases did not significantly increase retrieval performance.

6.2 Syntactic Phrases

A *syntactic phrase* may be selected by its occurrence frequency, the co-occurrence of its components, and/or upon the proximity of its components in the text, but there is always a syntactic relationship between the phrase components (Salton & McGill, 1983, p. 90 ff.; Croft et al., 1991; Strzalkowski, 1994; Strzalkowski et al., 1997). A syntactic phrase is a grammatical part of the sentence and is, at least in part, identified based upon linguistic criteria. The use of syntactic phrases is based on the assumption that words of a text that have a syntactic relationship often have a correlated semantic relationship (Smeaton & Sheridan, 1991). Syntactic phrase recognition has been popular for decades (an overview see Schwarz, 1990). In the following we describe the main recognition methods.

The simplest method uses a *machine-readable dictionary* or thesaurus that contains pre-coded phrasal terms according to various syntactic formats (cf. Evans, Ginther-Webster, Hart, Lefferts, & Monarch, 1991). Such dictionaries should encompass the many ways in which individual words can be combined to express the same concept, so their use is only practical in restricted subject domains.

A more realistic, but language-dependent method is based on the idea that content bearing phrases belong to certain grammatical classes or combinations of classes. The method has two steps: identification of the classes (*parts-of-speech*) of the words of the text and recognition of combinations of word classes in the text.

Word classes are defined by using a machine-readable dictionary of words with their classes or by using a *stochastic tagger.* A *stochastic tagger* (Dermatas & Kokkinakis, 1995) assigns part-of-speech tags to the words of a text based on the probability that the tag should be assigned to a word. This probability is computed taking into account the probability of a part-of-

speech tag for the specific word and the probability that a specific tag is appropriate for the particular context. The lexical and contextual probabilities are obtained from observing statistical regularities in example texts that are manually tagged with part-of-speech mark-ups.

There are two major ways for identifying *combinations of word classes* in texts: the use of syntactic templates and the parsing based on a context free grammar.

The former refers to matching patterns of adjacent classes against a library of *syntactic templates* (example of a template: adjective followed by a noun) (Dillon & Gray, 1983; Fuhr & Knorz, 1984).

In the latter way, a *context-free grammar*, which contains the rules of the allowable syntax of the sentences, is used to obtain for each sentence a parse that shows its syntactic structure (see chapter 6) (Salton 1968, p. 151 ff.; Metzler & Haas, 1989; Salton et al., 1990; Schwarz, 1990; Smeaton & Sheridan, 1991). The result of the parsing is captured with the formalism of a *dependency tree*, which reflects the logical predicate-argument structure of a sentence. The tree indicates dependencies between the phrase components of the sentence (e.g., head and modifier of a phrase). In this way, differences in meaning between phrases, such as “college junior” and “junior college”, are detected. Simple phrase structure grammars can be used to recognize many types of noun phrases and prepositional phrases that might constitute useful text identifiers. The simple grammars cannot account for all phrase structures and must be complemented with semantic knowledge in case of ambiguous syntactic structures (e.g., in the phrase “increasingly dangerous misadventures and accidents” the “accidents” are or are not “increasingly dangerous”) (Lewis, Croft, & Bhandaru, 1989). However, these problems do not prevent that currently there exists noun phrase recognition parsing algorithms that operate with low error rates.

Usually, a number of phrases are selected based upon their combination of grammatical classes, phrase frequency, and phrase weight (see below) (cf. Salton et al., 1990).

It must be noted that a *compound noun in Dutch* generally concatenates two (or more) words to create a single orthographic word. In case of compound nouns that were not split during a stemming procedure (see above), single Dutch words sometimes express very specific indexing concepts (e.g., “onroerendgoedmarkt” (“market of real estate”)).

Compared to single term indexing, Fagan (1989) found that syntactical phrase recognition only very slightly improved retrieval performance (cf. Strzalkowski, Ling, Perez-Carballo, 1998). A disadvantage of syntactic methods is their high demand of computer power, storage space, and program availability.

Part of the discouraging effect of the use of phrases in text retrieval is because they must be normalized to a standard form and they must be effectively selected. Normalization is discussed in the next section. The weighting of phrases for content representation is discussed further in this chapter. The solutions proposed primarily relate to noun phrases, because noun phrases are mostly selected from a text.

6.3 Normalization of Phrases

Indexing the text by considering phrases assumes that phrases refer to meaningful concepts. When in a retrieval environment a phrase appears in both query and document text, the two may refer to the same concept. This approach is limited by the fact that the phrase must appear in the same form in the document text and query in order for the concept to be matched (Lewis et al., 1989; Smeaton, 1992). However, this is rarely the case with phrasal terms. A same concept can be expressed using *different syntactic structures* (e.g., "a garden party" and "a party in the garden"), possibly combined with *lexical variations* in word use (e.g., "prenatal ultrasonic diagnosis" and "in utero sonographic diagnosis of the fetus") or with *morphological variants* (e.g., "vibrating over wavelets" and "wavelet vibrations"). Phrases may contain anaphors and ellipses. Correct mapping to a standard single phrase must take into account lexical, syntactic, and morphological variations and resolve anaphors and ellipses. In a retrieval environment, phrase normalization enhances the recall of a retrieval operation (Salton, 1986).

The following concerns important methods in phrase normalization.

1. A simple method is to use a machine-readable dictionary of phrase variants (e.g., Evans et al., 1991). Currently, such a dictionary is hand-built, which limits the method to restricted subject domains.
2. The *omission of function words* (e.g., propositions, determiners, pronouns) and possible *neglecting of the order of the remaining content words* forms another easy, but not always reliable, phrase normalization method (Dillon & Gray, 1983; Fagan, 1989).
3. A more secure method for recognition of syntactic variants is based on syntactical phrase recognition. It uses the output of a syntactic parse of a sentence and defines (meta)rules for equivalent phrases (Jacquemin & Royauté 1994; Strzalkowski et al., 1997; Tzoukermann, Klavans, & Jacquemin, 1997; cf. Sparck Jones & Tait, 1984). This approach may be combined with anaphoric resolution (see Grishman, 1986, p. 124 ff. and Lappin & Leass, 1994) and word stemming.

6.4 Recognition of Proper Names

A special case of phrase recognition in texts is the selection of *proper names* or *proper nouns* (Rau, 1992; Jacobs, 1993; Mani & MacMillan, 1996; Paik, Liddy, Yu, & McKenna, 1993; Strzalkowski et al., 1997). Indexing with important proper names is useful in many retrieval applications. Proper names regard names of persons, companies, institutions, product brands, locations, and currencies. There are two major ways for recognizing them.

1. The application of a *lexicon or machine-readable dictionary of names* requires an existing database of names, provided on an external basis (e.g., Hayes, 1994). Composing the database of names manually is only possible for applications with a narrow scope. The lexicon may provide name variants.
2. Because many proper names (e.g., companies) appear, disappear, or change, accurate identification requires recognizing new names. They are recognized by special rules that express the typical features of proper name phrases (e.g., capitalization) or the linguistic context (e.g., indicator words) in which the names ought to be found (Jacobs, 1993; Hayes, 1994; Cowie & Lehnert, 1996). Recognition is sometimes problematic (e.g., "van Otterloo & Coo").

Proper name recognition tools must cope with the many variants that occur. Variation in names concerns: suffix words (e.g., "Inc", "N.V."), prefix words (e.g., personal titles), other optional words (e.g., "van"), alternate words (e.g., "Intl Business Machines" and "International Business Machines"), alternate names (e.g., "IBM" and "Big Blue"), forenames (e.g., "Gerald Thijs", "G. Thijs", and "Thijs"), punctuation (e.g., "Sensotec N.V." and "Sensotec NV"), case sensitivity (e.g., "SigmaDelta" and "Sigmadelta"), and hyphenation (e.g., "Sigma Delta", "Sigma-Delta", and "SigmaDelta"). One way to resolve variants is by defining similarities between names based on shared letter sequences (*n*-grams) (cf. Pfeifer, Poersch, & Fuhr, 1996).

Another challenging problem is recognition of the semantic category of the proper names (e.g., identifying personal names, company names) (McDonald, 1996; Paik et al., 1993; Paik, Liddy, Yu, & McKenna, 1996). The category of a proper name can be extracted from the machine-readable dictionary, if available. Alternatively, the category can be detected by applying context heuristics that are developed from analysis of contexts in an example corpus.

7. INDEX TERM WEIGHTING

7.1 The General Process

The indexing process so far has generated a set of natural language index terms as the representation of the text. A term is typically a word, word stem, or phrase, Although the terms belong to the general class of content words, they are not equally important regarding the content of the text. An *importance indicator* or a *term weight* is associated with each index term. Term weighting is important to select good index terms for inclusion in the text representation or to better discriminate the index terms when matching a query in a retrieval environment (Salton & Yang, 1973; Buckley, 1993). Weighting enhances the precision of retrieval (Salton, 1986; Ro, 1988).

Many weighting functions have been proposed and tested (overviews see Sparck Jones, 1973; Salton, 1975a, p. 4 ff.; van Rijsbergen, 1979, p. 24 ff.; Noreault, McGill, & Koll, 1981; Salton & McGill, 1983, p. 59 ff. and p. 204 ff.; Ro, 1988; Salton & Buckley, 1988; Fuhr & Buckley, 1991; Tenopir, Ro, & Harter, 1991, p. 144-146). The following parameters play a role in the weight computation of an index term:

1. The index term itself: for instance, its syntactic class.
2. The text to be indexed: the parameters that describe the text: for instance, the length of the text and the number of different terms in the text.
3. The relation between the index term and the text to be indexed: for instance, the frequency of occurrence of the term in the text, the location of the term in the text, the relationship with other terms of the text, and the context of the term in the text.
4. The relation between the index term and the document (or another reference) corpus: for instance, its frequency of occurrence in this corpus.

Most weighting functions rely upon the distribution patterns of the terms in the text to be indexed and/or in a reference collection, and use statistics to compute the weights. The other parameters are less frequently employed. Only rarely, weights of index terms are determined based on *expert knowledge* on term importance (Sparck Jones, 1973).

The weight of an index term is usually a *numerical value.* Term weights have a value of zero or greater, or in case of normalized weights vary between zero and one, with values close to one indicating very important index terms and values close to zero very weak terms (Salton & Buckley, 1988). A zero value indicates that the term does not have any content value.

7.2 Classical Weighting Functions

The law of Zipf

It was Luhn (1957) who discovered that distribution patterns of words could give significant information about the property of being content bearing. He noted that high-frequency words tended to be common, non-content bearing words. He also recognized that one or two occurrences of a word in a relatively long text could not be taken significant in defining the subject matter. Earlier, Zipf (1949) plotted the logarithm of the frequency of a term in a body of texts against rank (highest frequency term has rank 1, second highest frequency term has rank 2, etc.). For a large body of text of "well-written English", the resulting curve is nearly a straight line. Thus, the *constant rank-frequency law of Zipf* describes the occurrence characteristics of the vocabulary, when the distinct words are arranged in decreasing order of their log frequency of occurrence:

$$\log (\text{frequency}) \cdot \text{rank} = \text{constant} \qquad (1)$$

This law expresses that the product of the logarithm of the frequency of each term and its rank is approximately constant. Other languages or other writing styles may be expressed by other non-linear functions. But, there is a relationship between the Zipfian curve and Luhn's concept of where the significant words are. Words with low significance are at both tails of the distribution. Therefor, Luhn suggested using the words in the middle of the frequency range. These findings are the basis of a number of classical weighting functions.

Term frequency

It is assumed that the degree of treatment of a subject in a text is reflected by the frequency of occurrence in the text of terms naming that concept. A writer normally repeats certain words as he or she advances or varies the arguments and as he or she elaborates on an aspect of the subject. This means of emphasis is taken as indicator of significance. A content term that occurs frequently in a text is more important in the text than an infrequent term. The frequency of occurrence of a content word is used to indicate term importance for content representation (Luhn, 1957; Baxendale, 1958; Salton, 1975a, p. 4 ff.; Salton & McGill, 1983, p. 59 ff.; Salton, 1989, p. 279).

The term frequency (*tf*) measures the frequency of occurrence of an index term in the document text (Salton & Buckley, 1988):

tf_i = frequency of occurrence of the index term i in the text. (2)

The occurrence of a rare term in a short text is more significant than its occurrence in a long text. The logarithmic term frequency reduces the importance of the raw term frequency in those collections with wide varying text length (cf. length normalization below) (Sparck Jones, 1973; Salton & Buckley, 1988; Lee, 1995):

$\log(tf_i)$ = common logarithm of frequency of occurrence of index term i in the text (3)

$\ln(tf_i)$ = natural logarithm of frequency of occurrence of index term i in the text. (4)

Index terms with a high term frequency are good at representing text content, especially in long texts and in texts containing many significant or technical terms. For short texts, term frequency information is negligible (most of the terms occur once or twice) or even misleading. Anaphoric constructs and synonyms in the text hide the true term frequency (Bonzi & Liddy, 1989; Smeaton, 1992). It is assumed that high frequency content-bearing terms represent the main topics of the text. When an index term occurs with a frequency higher than one would expect in a certain passage of the text, it possibly represents a subtopic of the text (Hearst & Plaunt, 1993).

Inverse document frequency

After elimination of stopwords, a text still contains many common words that are poor indicators of its content. Common words tend to occur in numerous texts in a collection and often seem randomly distributed over all texts. The more texts a term occurs in, the less important it may be. For instance, the term “computer” is not a good index term for a document collection in computing, no matter what its frequency of occurrence in a text of the collection. The more rarely a term occurs in individual texts the more discriminating that term is. Therefor, the weight of a term should be inversely related to the number of document texts in which the term occurs, or to the document frequency of the term (Sparck Jones, 1972; Salton & Yang, 1973; Salton, 1975a, p. 4 ff.; Salton & McGill, 1983, p. 63; Salton, 1989, p. 279 ff.; Greiff, 1998). An inverse document frequency factor (*idf* factor), is commonly used to incorporate this effect. The logarithm decreases the effect of the inverse document frequency factor. *The inverse document*

frequency (*idf*) weight is commonly computed as (Sparck Jones, 1973; Salton & Buckley, 1988; Lee, 1995):

$$\log\left(\frac{N}{n_i}\right) \tag{5}$$

where

log = common logarithm (an alternative is ln = natural logarithm)

N = number of documents in the reference collection

n_i = number of documents in the reference collection having index term *i*.

An inverse document frequency weight is *collection dependent*. It is usually obtained from a collection analysis prior to the actual indexing of the documents and is based on the distribution of the term in a reference collection. The reference collection is customarily the complete text corpus to be indexed. It may also be a general corpus that reflects a broad range of texts (e.g., the Brown corpus in English) (cf. Evans et al., 1991). When the reference collection changes over time, the weight of an index term should be recomputed each time a document is added to or deleted from the collection. This is not only unpractical, but results in an unstable text representation. So, the use of the inverse document frequency factor based on a changing reference collection is discouraged (Salton & Buckley, 1988). Other types of reference collections are possible. For instance, Hearst and Plaunt (1993) consider the complete text of a document as the reference frame for computing the weight of index terms of small text segments (3-5 lines) in order to discriminate the subtopics of these segments.

The inverse document frequency factor is important in identifying content bearing index terms in texts (Sparck Jones, 1973). Sometimes, index terms with a low inverse document frequency value are eliminated as stopwords (e.g., Smeaton, O'Donnell, & Kelledy, 1995).

Product of the term and the inverse document frequency

In judging the value of a term for purposes of content representation, two different statistical criteria come into consideration. A term appearing often in the text is assumed to carry more importance for content representation than a rarely occurring term. On the other hand, if that same term occurs as well in many other documents of the collection, the term is possibly not as valuable as other terms that occur rarely in the remaining documents. This suggests that the specificity of a given term as applied to a given text can be measured by a combination of its frequency of occurrence inside that text

(the term frequency or *tf*) and an inverse function of the number of documents in the collection to which it is assigned (the inverse document frequency or *idf*). The best terms will be those occurring frequently inside the text, but rarely in the other texts of the document collection. These findings are the basis for a very popular term weighting function that determines *the product of the term frequency and the inverse document frequency* (*tf* x *idf*) of the index term (Sparck Jones, 1973; Salton, 1975a, p. 26 ff,; Salton & Buckley, 1988; Salton, 1989, p. 280 ff.; Harman, 1986 cited in Harman, 1992a). Usually, the product of the raw term frequency (2) and the common logarithm of the inverse document frequency (5) is computed:

$$tf_i \cdot \log\left(\frac{N}{n_i}\right) \tag{6}$$

Length normalization

Document texts have different sizes. Long and verbose texts usually use the same terms repeatedly. As a result, the term frequency factors are large for long texts and small for short ones obscuring the real term importance. Also, long texts have numerous different terms. This increases the number of word matches between a query and a long text, increasing the chances of retrieval over shorter texts. To compensate for these effects, variations in length can be normalized. *Length normalization* is usually incorporated in weighting functions[2] and it mostly normalizes the term frequency factor in a weighting function. The following describes the most important length normalization functions.

The term frequency of an index term *i* is sometimes normalized by dividing the term frequency (2) by the maximum frequency that a term occurs in the text:

$$\frac{tf_i}{\max tf_j} \tag{7}$$

where

tf_j = term frequency of an index term j in the text

j = 1 .. n (n = number of distinct index terms in the text).

The result of the above normalization is a term frequency weight that lies between 0 and 1. In a popular variant the normalized term frequency of (7) is weighted by 0.5 to decrease the difference in weights of terms that occur infrequently and terms that occur frequently. The weighted term frequency is further altered to lie between 0.5 and 1 (addition of 0.5). This variant is

called the *augmented normalized term frequency* (Salton & Buckley, 1988; Lee, 1995):

$$0.5 + 0.5\left(\frac{tf_i}{\max tf_j}\right) \tag{8}$$

A common way of length normalization is the cosine normalization where each term weight is divided by a factor representing Euclidean vector length (Salton & Buckley, 1988). The length of the vector is computed with all distinct indexable words. When the weight of the index term *i* is computed with the term frequency (*tf*) (2), the normalized term weight of index term *i* is:

$$\frac{tf_i}{\sqrt{\sum (tf_j)^2}} \tag{9}$$

where

tf_j = term frequency of an index term *j*

j = 1 .. *n* (*n* = number of distinct index terms in the text).

Cosine length normalization can be applied to other weighting functions, such as the product of the term frequency and the inverse document frequency (*tf* x *idf*) (6), which yields the normalized term weight for index term *i* (cf. 9):

$$\frac{tf_i \cdot \log\left(\frac{N}{n_i}\right)}{\sqrt{\sum\left(tf_j \cdot \log\left(\frac{N}{n_j}\right)\right)^2}} \tag{10}$$

Length normalization is beneficial for certain texts. It has been proven successful for indexing a document collection with texts of varying length (Sparck Jones, 1973; Salton & Buckley, 1988), especially when long texts are the result of *verbosity.*

Long texts have other causes than solely verbosity. One of them is the presence of *multiple topics.* In this case, the cosine normalization (9 and 10) causes the weight of a topic term to be decreased by the weight of non-relevant terms, i.e., terms that discuss the other topics. In a retrieval

environment, this situation decreases the changes of retrieving documents that deal with multiple topics, when only one of the topics is specified in the query (Lee, 1995). The augmented normalized term frequency (8) alleviates this effect. This is because, the normalizing factor of this method, namely the maximum frequency of term occurrence in the text usually has a modest value when the text deals with multiple topics. Another reason for long texts is that they contain *much information about a specific topic.* In a retrieval environment long documents are sometimes preferred over shorter ones that treat the same topic (Singhal, Salton, Mitra, & Buckley, 1996). Length normalization is a way of penalizing the term weights for longer documents, thereby reducing, if not removing completely, the advantage of long documents in retrieval (Strzalkowski, 1994). *Pivoted length normalization* increases or decreases the impact of a length normalization factor (Singhal, Buckley, & Mitra, 1996). Initial training queries retrieve an initial set of documents and the probabilities of relevance and of retrieval are plotted against text length. Pivoted normalization makes the normalization function weaker or stronger by reducing the deviation in the retrieval probabilities from the likelihood of relevance.

Term discrimination value

The *term discrimination model* (Salton, Yang, & Yu, 1975; Salton & McGill, 1983, p. 66 ff.; Salton, 1989, p. 281 ff.) assumes that the most useful terms for content identification of natural language texts are those capable of distinguishing the documents of a collection from each other. The term discrimination value measures the degree to which the use of the term will help distinguishing the documents from each other. For this purpose, the concept of *connectivity* is used. Bad index terms are the ones that increase the degree of connectivity between texts, while good index terms decrease it. The term discrimination value of an index term is computed as the difference in connectivity between the texts, before and after adding the index term. The simplest way to compute the degree of connectivity is by taking the average of all mutual similarities between the text pairs in the collection. Similarities between the texts are obtained with similarity functions applied upon their term vectors (cf. Jones & Furnas, 1987).

The term discrimination value is *collection dependent.* The value is comparable with the inverse document frequency weight and may replace the latter in a *tf* x *idf* weighting function (6) (Salton, Yang, & Yu, 1975). However, while very frequent terms tend to have low weights for either function, discrimination values for medium frequency terms tend to be higher than for low frequency terms (Sparck Jones, 1973). The term

discrimination model has been criticized, because it especially discriminates a document from all other documents of the collection (Salton, 1989, p. 284). It is possible that many other relevant documents regarding the topic expressed by an index term are present in the collection.

Term relevance weights

A *term relevance weight* of an index term is learned based upon its probability of occurrence in relevant and non-relevant documents (Maron & Kuhns, 1960; Salton, 1989, p. 284 ff.). The relevant and non-relevant set are assumed to be representative for the complete corpus. Commonly, term relevance weights are computed on the basis of relevance information from a number of queries formulated with the index term. The term relevance weights are based on term occurrence characteristics in the relevant and non-relevant texts. For example, terms occurring mostly in texts identified as relevant to the query receive higher weights than terms occurring in the non-relevant texts. A number of different relevance weighting functions have been formulated (Bookstein & Swanson, 1975; Robertson & Sparck Jones, 1976; Sparck Jones, 1979; Salton, 1989, p. 284 ff.; Fuhr & Buckley, 1991). A preferred function for the weight of an index term i is (Robertson & Sparck Jones, 1976; Sparck Jones, 1979):

$$\log \frac{\left(\frac{r_i}{R - r_i}\right)}{\left(\frac{n_i - r_i}{N - n_i - R + r_i}\right)} \qquad (11)$$

where

N = the number of texts in the training set

R = the number of relevant texts for the query

n_i = the number of texts having index term i

r_i = the number of relevant texts having index term i.

In real applications, it is difficult to have enough relevance information for each index term available in order to estimate the required probabilities (cf. Croft & Harper, 1979; Robertson & Walker, 1997).

Phrase weighting

It has been shown that phrases give potentially better coverage of text content than single-word terms. When selecting phrases from a text, not all

phrases equally define its content. A text can contain very specific concepts that are of no importance to include in its representation. *Phrase weighting* (including proper name phrase weighting) helps in deciding which phrases to include in the representation. Phrase weighting also contributes to a better discrimination of phrasal terms when matching query and text representations in a retrieval process.

Because of their lower frequency and different distribution characteristics, weighting of phrases can differ from single-word weighting (Fuhr, 1992; Lewis & Sparck Jones, 1996). However, the methods currently in use employ in one way or another classical weighting functions for single words, It is generally agreed that phrase weighting needs further investigation (Fagan, 1989; Croft et al., 1991; Buckley, 1993; Strzalkowski et al., 1997).

When computing a phrase weight, a phrase can be considered as a separate concept or as a set of words (Croft et al., 1991).

1. When the phrase is considered as a *separate concept,* its weight is independent of the weight of its composing components. This weight can be proportional with the number of times the phrase occurs in the text *(term frequency tf*) (2) and/or inversely proportional with the number of document texts in which the phrase occurs (*inverse document frequency idf*) (5) (Dillon & Gray, 1983; Croft et al., 1991; Strzalkowski, 1994). In order to obtain accurate weights, such a strategy requires a correct normalization of the phrases to a standard form and a resolution of anaphors[3] (Smeaton, 1986).
2. The weight of a phrase can be the *combination of the weights* of its *composing single words.* Then, the weight is computed as the average weight of the components (Salton, Yang, & Yu, 1975; Fagan, 1989; Croft et al., 1991; Evans et al., 1991), as the product of the component weights (Croft et al., 1991), or as the highest weight amongst the component weights (Croft et al., 1991). The weight of a phrase component is usually computed as the product of the *term frequency* (*tf*) (2) and the *inverse document frequency* (*idf*) (5) of the individual word. Although the weight of a single component may influence the weight of the phrase, this strategy makes it possible that a phrase weight usually does not differ strongly from the weights of its components (Fagan, 1989).
3. Jones, Gassie, and Radhakrishnan (1990) employ a combined approach and weigh phrases proportional to the frequency of occurrence of the complete phrase and to the frequency of occurrence of its composing words.

8. ALTERNATIVE PROCEDURES FOR SELECTING INDEX TERMS

8.1 The Multiple Poisson (*n*P) Model of Word Distribution

The Multiple Poisson (*n*P) model of word distribution has been proposed as a *statistical model of word distribution* in large collections of full-text documents. Words are assumed being distributed at random. The process of generation of texts can be viewed as a *stochastic process* where the texts are created by randomly selecting text tokens. The Poisson distribution is a discrete random distribution that can be used to model a variety of random phenomena. The number of words that can be used to create a text is very large and the probability of being selected for each word is small, so the process of text generation can be seen as a Poisson process. Probabilistic theory hypotheses that compliance with or deviation from a Poisson process of the distribution of a word could be exploited for indexing purposes.

In a *first model* (Bookstein & Swanson, 1974), it is assumed that *common words*, which do not indicate content, are likely to be distributed at random in a document collection. Because they exhibit the same occurrence properties in all the texts of a collection, a *single Poisson distribution* can characterize them. Then, good index words are the words the distribution of which deviates significantly from the expected single Poisson distribution. Margulis (1 992) tested whether stopwords are Poisson distributed and found that in small collections stopwords are single Poisson distributed, but this was unlikely to be the case in large collections because of interfering noise.

Words that reflect content tend to be clustered in a subset of the document corpus and a single Poisson distribution or deviation from such a distribution cannot be used to represent their properties across the documents of a collection. In their second model, Bookstein and Swanson (1974) suggest that in the stochastic process, when the texts are created by randomly selecting text tokens, each text token i has a certain probability of being selected in document text D_m. This probability depends upon the extent of *topic coverage* associated with i in D_m. A text usually covers to a large extent one or a few main topics, to a lesser extent it discusses other topics. A document collection can be divided into subsets of texts reflecting the extent of coverage of a certain topic. These subsets are referred to as “levels” or “classes” of topic coverage. As seen above, it is assumed that the *frequency of occurrence* of a specific term in a particular text depends on the extent to which this text is related to the topic associated with the term. Thus, the

extent of topic coverage represented by a specific term in the text can be approximated by its frequency of occurrence.

The number of topic words is very large and the probability of being selected is small, so the process of topic term generation is a Poisson process. But, the mean of this Poisson process depends upon the degree of topic coverage associated with the topic term. The document collection can be broken down into subclasses regarding the topic coverage of a specific term, and the assumption is made that a different Poisson distribution applies to the given term in each subclass with different parameters. The distribution of the text token *i* within *each class* C_j is governed by a *single Poisson process* with a mean of λ_j. This is often computed as the average number of occurrences of the text token *i* per text in this class and represents the extent of the topic associated with *i*.

The occurrence of a certain word across all texts is than divided according to a *Multiple Poisson* (*n*P) distribution, of which the number of components is equal to the number of classes. A Multiple Poisson distribution is a mixture of Poisson distributions with different means (λ_j). Thus, the distribution of a certain text term *i* in texts within the whole collection is governed by the sum of Poisson distributions, one for each class of topic coverage. The frequency of occurrence of a text word is then described by a sum of Poisson distributions. Each summand in this sum is an independent single Poisson distribution that describes the frequency of occurrence within a subset of texts that belong to the same level of topic coverage related to the text term. The probability that a randomly chosen document text D_m contains *k* occurrences of a certain term *i* is given by:

$$P(dfreq(D_m, i) = k) = \sum_j \pi j \frac{\lambda_j^k}{k!} e^{-\lambda_j} \tag{12}$$

where

j = class of topic coverage related to the term i

λj = average extent of topic coverage related to the term i within the class Cj

πj = probability that the text belongs to a class Cj and given $\Sigma_j \pi_j = 1$.

The validity of the Multiple Poisson (*n*P) model has been tested for single words (Bookstein & Swanson, 1974; Harter, 1975a, 1975b; Losee, 1988; Srinivasan, 1990). The study of Margulis (1992) indicates that over 70% of frequently occurring words and word stems indeed behave according to the Multiple Poisson model. The proportion of words that are Multiple Poisson distributed depends on the collection size, text length, and the frequency of

individual words. Most of the words are distributed according to the mixture of relatively few single Poisson distributions (two, three or four).

For *indexing purposes,* it is important to compute for each term the extent of topic coverage in order to select the term as index term or to appropriately weigh the term. So, the ultimate aim of the Multiple Poisson (*n*P) model of word distribution is that the division of texts in classes gives insight into the content of the texts based on a number of word occurrences. Assuming that terms in a body of text are generated by a Poisson process, allows measuring the probability that a text has a given number of occurrences given an average frequency of occurrences of the term in a class about the topic in a reference or example collection. The probability that a text with *k* occurrences of the index term *i* belongs to a certain class of topic coverage (*Cx*) with a mean of λ_x regarding the use of index *i* can be computed by (cf. 12) (cf. van Rijsbergen, 1979, p. 28 ff.):

$$\frac{\pi_x \frac{\lambda_x^k}{k!} e^{-\lambda x}}{\sum_j \pi j \frac{\lambda_j^k}{k!} e^{-\lambda}}$$

(13)

For each class of topic coverage regarding index term *i,* this probability can be computed and used as a criterion for class membership (and consequently as a criterion for selection of the index term) or used as a probabilistic term weight. The difficulty in using this approach lies in the estimation of the parameters, especially in estimating the means of each Poisson distribution. A common technique estimates the parameters of a two-Poisson distribution for each term directly from the distribution of within-text frequencies in the class of example texts that is about the topic term and in the class of example texts that does not bear upon the topic term (Robertson, van Rijsbergen, & Porter, 1981). Estimation of the parameters needs further research (cf. Losee, 1988; Robertson &Walker, 1994).

8.2 The Role of Discourse Structure

Knowledge about *discourse structures* and their *signaling linguistic phenomena* can help in selecting terms from a text that are reflective of its content (Hahn, 1989; Lewis & Sparck Jones, 1996). The idea can be traced back to Luhn (1957). There are timid attempts to incorporate knowledge about discourse structures into text indexing. Dennis (1967) determines the importance of a word based upon its frequency of occurrence within a text paragraph and across preceding and succeeding paragraphs. The tendency of

occurrences of a word to clump is still considered useful in selecting terms (Bookstein, Klein, & Raita, 1998). Index term selection and weighting can be determined by the structural position of the term in the text (e.g., within title, within summary, in a first paragraph) (Bernstein & Williamson, 1984; Jonák, 1984; Wade, Willett, & Bawden, 1989; Liddy & Myaeng, 1993; Wilkinson, 1994; Burnett, Fisher, & Jones, 1996; Burger, Aberdeen, & Palmer, 1997; Fitzpatrick, Dent, & Promhouse, 1997). There is also much research into structural decomposition of texts according to different themes (Salton & Buckley, 1991; Hearst & Plaunt, 1993; Salton, Allan, Buckley, & Singhal, 1994; Salton, Singhal, Mitra, & Buckley, 1997), which might be useful for identifying important topic terms in texts.

9. SELECTION OF NATURAL LANGUAGE INDEX TERMS: ACCOMPLISHMENTS AND PROBLEMS

Selecting natural language index terms from texts is a simple and often computationally efficient way to index texts and is therefor used to index large and heterogeneous text collections (e.g., indexing documents on the Internet cf. Szuprowicz, 1997). The *Text REtrieval Conferences* (*TREC*) describe recall and precision results when information retrieval is based upon natural language index terms (see Harman, 1993, 1994, 1995, 1996; Voorhees & Harman, 1997, 1998, 1999). In the TREC experiments a static, large collection of documents is searched for specific topics by different systems. The following results are only meant to give a rough idea of retrieval effectiveness. In retrieval, recall and precision can be computed at a specific cut-off number of highest-ranking documents. In TREC experiments, at a cut-off upon which half of the retrieved documents is relevant (recall of 0.5), precision is usually below 0.4. Performance of search engines on the Internet is even worse (Gordon & Pathak, 1999). For different search engines, at a cut-off of twenty documents average recall and precision are below 0.16 and 0.4 respectively. At a cut-off of 200 documents average recall and precision are below 0.25 and 0.1 respectively. As seen in chapter 1, performance might be improved by using better indexing techniques. This chapter learns that selection of natural language terms can be refined.

1. Not all words of a text are good index terms and their *discrimination* is mandatory. Excluding stopwords and weighting terms are important. However, term weighting, especially phrase weighting, can be improved (cf. Strzalkowski et al., 1997). The relation between term distributions

and the topics of a text needs further research. It might be that the thematic structure of sentences or of whole texts yields good cues for topic term selection.

2. Single words are sometimes *too general* in meaning to convey text content. In these cases, text phrases are better indicators of text content. On the other hand, some index terms are *too specific* to represent text content (e.g., different morphological variants of words or syntactical variants of phrases that express the same concept). Stemming procedures and normalization of phrases alleviate this problem. Reducing phrases that express the same concept to a standard form needs further research.

10. CONCLUSIONS

Many of the techniques for selecting natural language index terms from texts rely upon simple assumptions about distribution patterns of individual words. In some cases the methods bear upon linguistic knowledge regarding a *micro level of text description*, i.e., bearing upon the vocabulary, syntax, and semantics of the individual sentences, clauses, and phrases. The linguistic knowledge is involved in stemming procedures, phrase recognition, and phrase normalization. The existing techniques were originally developed to index heterogeneous document collections, which explains the rather shallow approach. There is a growing interest to incorporate knowledge regarding a *macro level of text description* into the indexing systems. This knowledge can not only be incorporated as heuristics in the search for good terms, but can also be the basis of probabilistic term distributions that are useful in indexing.

There are of course the problems of indexing with natural language terms discussed in chapter 3 including synonymy, homonymy, and polysemy and the set of terms being an unordered set of phrases or individual words. In the next chapters of this part, alternative text indexing and abstracting techniques are described that alleviate these problems.

---

[1] The Dice coefficient (Jones & Furnas, 1987): $2C / (A + B)$, with A = number of unique bigrams in the first word, B = number of unique bigrams in the second word, and C = number of unique bigrams shared by A and B.

[2] Length normalization may be part of matching query and document, when similarity functions incorporate a length normalization factor (eg., division by the product of the Euclidean lengths of the vectors to be compared in the cosine function) (Jones & Furnas, 1987).

[3] Strzalkowski (1994) multiplies the product of the *term frequency* (*tf*) and *inverse document frequency* (*idf*) with a constant to increase the weight of phrasal terms in order to account for unresolved anaphors.

Chapter 5

AUTOMATIC INDEXING: THE ASSIGNMENT OF CONTROLLED LANGUAGE INDEX TERMS

1. INTRODUCTION

The concepts discussed in a text can be expressed in many different ways. As it is demonstrated in the chapter 3, there are many problems inherent to the use of natural language index terms for representing a text‘s content, most importantly their semantic ambiguity and their difficulty in using them in generic searches. The use of controlled language index terms can to a large degree solve these problems. The controlled language index terms or *descriptors* concern *terms from a subject thesaurus*, broad *subject headings*, and *classification codes* (Harter, 1986, p. 40 ff.). Assignment of subject or classification codes is also referred to as *text categorization*.

Automatic assignment of controlled language index terms is an idea that already goes back to Luhn (1957). The assignment is based upon knowledge about typical text patterns (e.g., occurrences of typical words or combinations hereof) and their relationship with the concept represented by the index term. This knowledge is often manually acquired and implemented. Currently, there is a large interest to automate the knowledge acquisition step, which not only reduces the cost of implementation, but also more importantly, gives opportunities to broaden the subject domain and text typology of the application. Research efforts attempt to automatically construct thesauri and acquire the knowledge of textual patterns involved in text categorization.

This chapter outlines the most important techniques. Evaluation of the index terms is shortly described. We discuss the assignment of terms of a thesaurus and the automatic construction of thesauri. An important part of this chapter regards the assignment of subject and classification codes. Much emphasis is on the techniques that automatically learn the text patterns involved in text categorization. We discuss statistical approaches, learning of rules and trees, and neural networks. We finish the chapter by specifying the problems and accomplishments of the described techniques.

2. A NOTE ABOUT EVALUATION

Thesaurus terms replace the natural language index terms of a text. They are usually evaluated by measuring the *retrieval effectiveness in terms of recall and precision.* This is a form of extrinsic evaluation (cf. Sparck Jones & Galliers, 1996, p. 19 ff.). We refer to chapter 4 p. 78 for the definitions of recall and precision in this kind of evaluation.

The effectiveness of automatic assignment of *subject headings and classification codes* is more directly computed by comparing the results of the automatic assignment with the manual assignments by an expert. This is a form of intrinsic evaluation (cf. Sparck Jones & Galliers, 1996, p. 19 ff.). The text categorization is seen as a binary decision: A document text belongs or does not belong to a specific class or category. Table 1 summarizes the relationships between the system classifications and the expert judgments for the class C_k (Lewis, 1995).

$$\text{Recall} = a / (a + c) \quad (1)$$

$$\text{Precision} = a / (a + b) \quad (2)$$

$$\text{Fallout} = b / (b + d) \quad (3)$$

Recall is the proportion of class members that the system assigns to the class. *Precision* is the proportion of members assigned to the class that really are class members. *Fallout* computes the proportion of incorrect class members given the number of incorrect class members that the system could generate. Ideally, recall and precision are close to 1 and fallout is close to 0.

When comparing two classifiers, it is desirable to have a single measure of effectiveness. The *error rate,* which is also based on the above contingency table, takes into account both errors of commission (b) and errors of omission (c) (Lewis, 1995):

$$\text{error rate} = (b + c) / n \quad (4)$$

Table 1. Contingency table of classification decisions.

	Expert says yes	Expert says no	
System says yes	a	b	$a + b = k$
System says no	c	d	$c + d = n - k$
	$a + c = r$	$b + d = n - r$	$a + b + c + d = n$

where

n = number of texts in the test base

k = number of texts classified as relevant for the class C_k by the system

r = number of texts classified as relevant for the class C_k by the expert.

The *E-measure* combines recall and precision of the categorization operation (cf. van Rijsbergen, 1979, p. 174-175):

$$E = 1 - \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{5}$$

where

P = precision

R = recall

β = a factor that indicates the relative importance of recall and precision.

Ideally, the error rate and the E-measure are close to 0. To get a single measure of effectiveness where higher values (ideally 1) correspond to better effectiveness, and where recall and precision rate are of equal importance (ß = 1), the *F-measure* is defined in terms of the E-measure (5) (Lewis & Gale, 1994; Lewis, 1995):

$$F_{\beta = 1} = 1 - E_{\beta = 1} \tag{6}$$

Similarly, *accuracy* is defined in terms of the error rate (4):

$$\text{accuracy} = 1 - \text{error rate} = (a + d) / n \tag{7}$$

When multiple categories can be assigned to the document corpus, the results of the above measurements for each category can be averaged over categories (*macro-averaging*) or over all binary categorization decisions (*micro-averaging*) (Lewis, 1992a). The latter way of averaging provokes that categories with many texts have a larger impact upon the results.

3. THESAURUS TERMS

A first and common form of vocabulary control is the assignment of index terms as listed and described in a *thesaurus* (Harter, 1986, p. 42 ff.). The thesaurus offers a precise vocabulary to describe a document text. The original terms of the text are transformed to more uniform naming or more general concepts. A thesaurus for automatic indexing has the form of a *machine-readable dictionary* (*MRD*).

A thesaurus provides a grouping or classification of the terms used in a given topic area into classes known as *thesaurus classes*. The terms of a thesaurus class have a certain semantic relatedness due to their inherent meanings (Salton, 1975b, p. 461 ff.). Each class has a representative term, called a thesaurus class term. The thesaurus is used to replace a text's term by its *thesaurus class term*. Class membership can be weighted (Mc Cune, Tong, Dean, & Shapiro, 1985).

A thesaurus portraits the semantic relationships that hold between the terms when they refer to different aspects of a common concept or domain (Fox, 1980; Wang & Vandendorpe, 1985; Fagan, 1989). The main relationships are the ones that define synonyms or that broaden or narrow the meaning of a term. Other kinds of semantic relationships are possible.

Thesaurus classes have a similar function as *ontologies* used in natural language processing. Whereas philosophical work on ontology traditionally concerns questions about the nature of being and existence, in artificial intelligence communities ontologies refer to the general organizations of concepts and entities found in knowledge representations, which are sharable and reusable across knowledge bases (Bateman, 1995). In natural language processing, ontologies have been primarily used for modeling the semantics of lexical items (Dahlgren, 1995).

3.1 The Function of Thesaurus Terms

The main function of a thesaurus is to *generalize* or *make uniform* terms that have a related meaning, but unrelated surface forms, into more general and uniform index terms. More specifically, a thesaurus has the following functions (see also Miller, 1997).

1. A first important function is to control the *synonym* problem of natural language (Salton, 1975b, p. 461). Synonym words (e.g., "pests" and "vermin") can be handled by word substitution. The thesaurus puts words that are synonyms and are intersubstitutable into *equivalence classes*. If natural language contains several terms that might be used to represent

the same or nearly the same concept, the thesaurus usually guides the choice of vocabulary toward a single valid term. Even in restricted subject domains, a synonym list can become quite large. Substitution with true synonyms can be handled effectively, but there is the problem of near synonyms. A thesaurus can also be used to generalize *morphological* and *syntactical variants* of index terms when stemming terms or normalizing phrases (see chapter 4).

2. In case the thesaurus offers a hierarchical relationship between the words that it contains, it can be employed to *broaden terms* (Salton, 1975b, p. 461). Then, a term extracted from the text is replaced by a broader thesaurus class term. Such broad index terms are useful for generic searches and routing tasks. Occasionally, a thesaurus can be used to *narrow terms*.
3. In natural language many words have more than one semantic meaning or sense. So, a thesaurus may contain word senses from which the meaning of a polysemous or homonymous word may be chosen (Voorhees, 1994). For indexing text, the use of such a thesaurus supposes a procedure for identifying the meaning. Techniques for *word sense disambiguation* (see Krovetz & Croft, 1992; Guthrie, Pustejovsky, Wilks, & Slator, 1996) include the application of knowledge of the syntactic class of the word to be indexed (e.g., noun) and of domain knowledge that relates a word class to a word meaning. When a word sense is not or not solely determined by its syntactic class, selecting the correct word sense or the most probable word sense is only feasible by considering the context in which the term occurs. A word's context varies from the local context (e.g., words in the same sentence or surrounding sentences) and the complete text in which the word occurs, to the complete corpus (e.g., to disambiguate word senses in short texts). How best to characterize the contexts associated with word senses for automated word sense disambiguation remains an open question. When people disambiguate word senses in reading, they seem to make more use of local context: the exact sequence of words immediately preceding and following the polysemous word (Miller, 1995). Machine-readable dictionaries employed in word sense disambiguation contain for each sense of each word a short textual description. This description can be used in disambiguation, for instance, by searching for occurrences of words from the description in the document (Lesk, 1986 cited in Krovetz & Croft, 1992). Alternatively, categories can be defined representing the different senses of a word (Voorhees, 1994). Then, the number of words in the text that have senses that belong to a given category is counted. The senses that correspond to the category with the largest counts are selected to be

the intended senses of the ambiguous words. In restricted subject domains contextual rules can be implemented to disambiguate word senses (Krovetz & Croft, 1992). We refer to the special issue of *Computational Linguistics* on word sense disambiguation (24 (1), 1998).

Thesaurus class terms have been effective for indexing document texts. They can replace the natural language index terms extracted from a text. Alternatively, they can complement the natural language index terms of a text representation. This is in analogy to the use of a thesaurus to expand terms of a query with related terms in a retrieval system (van Rijsbergen, 1979, p. 31 ff.; Salton & Lesk, 1971; Fox, 1980; Gauch & Smith, 1991). Thesaurus class terms enhance the recall of a retrieval operation. A thesaurus is useful to index a text by *word senses*. Indexing by word senses increases the precision of a retrieval operation (Krovetz & Croft, 1992). Especially in restricted subject domains where the community of scholars and scientists working in the discipline shares word meanings, thesaurus class terms are very useful index terms. However, for heterogeneous text collections, more must become known about the desired form and content of thesauri and about the processes of word sense disambiguation that can be automated (Smeaton, 1992; Schütze & Pedersen, 1994).

3.2 Thesaurus Construction and Maintenance

An important problem with the use of thesauri is their construction and maintenance. Thesauri are usually manually constructed. Sometimes, on-line versions of existing published dictionaries are available. Additionally, there are efforts to automatically or semi-automatically build thesauri.

Building a thesaurus *manually* or *intellectually* is a time-consuming and costly task. It is usually constructed by a committee of experts who review the subject matter and propose reasonable class arrangements (Salton, 1989, p. 301). The thesaurus classes cover restricted topics of specified scope and they collectively cover the complete subject area evenly. Hand-built thesauri are often only confined to restricted subject domains and are usually not employable outside the collections.

However, in the past many dictionaries have been built manually. Dictionary entries evolved for the convenience of human readers, and not for being used by machine. But, this is changing. The thesaurus becomes an online version of a *semantically coded dictionary* (see Guthrie et al., 1996 for an overview). Roget already in 1946 used a procedure for compiling a thesaurus of English words (cited in Luhn, 1957). He created categories of words that had a family resemblance on a conceptual level and arrived at

approximately 1000 of these categories. Also, in the *Longman's Dictionary Of Contemporary English* (*LDOCE*) (published in 1981) lexicographers supplemented the machine-readable version with codes that give the semantic category of a word. LDOCE can be used to disambiguate word senses. Parsers have been developed that analyze the definition texts of LDOCE (see Boguraev & Briscoe, 1989). Networks of noun senses for both the LDOCE and the Dutch Van Dale Dictionary have been created using a technique for disambiguation that combines information from both dictionaries with information from the Van Dale bilingual Dutch-English dictionary (Guthrie et al., 1996). Another example of an on-line dictionary is *WordNet*, a lexical database for English developed at Princeton University, NJ (Miller, 1990, 1995). It contains words, word senses, syntactic word classes, and important semantic relations between words. A current goal of WordNet is developing tools for determining a word sense based on the context in which a word is used. An important on-line lexical database for Dutch is CELEX, created by the Centre for Lexical Information at the Katholieke Universiteit Nijmegen. The availability of large on-line thesauri increases the applicability of assigning thesaurus class terms when indexing (Fox, Nutter, Ahlswede, Evens, & Markowitz, 1988; Liddy & Myaeng, 1993; Liddy & Paik, 1993; Liddy, Paik, & Yu, 1994). A generic on-line published thesaurus is often restricted to common usage of words. When used for technical domains, which have their own terminology, it will have serious coverage gaps. Specialized dictionaries that cover the important terms and concepts of their disciplines may expand the coverage of a standard dictionary.

One major disadvantage inherent to the use of any thesaurus is the necessity to maintain it. New thesaurus classes of interest emerge and the thesaurus needs to accommodate for collection growth. Especially, in some disciplines where the vocabulary changes rapidly (e.g., computer science) maintenance of the thesaurus is important. The cost of implementing and maintaining an on-line thesaurus, as well as the need for collection-specific thesauri, incites research to *build thesauri automatically or semi-automatically.* Research focuses in discovering related words directly from the contents of a textual database. This research dates back to Dennis (1967), to Sparck Jones' work on term classification (1970, 1971), to Salton's work on automatic thesaurus construction and query expansion (1968, 1980), and to van Rijsbergen's work on term co-occurrence (van Rijsbergen, Harper, & Porter, 1981). Generally, thesauri generated automatically attempt to identify semantic relationships between words based on statistical and syntactic patterns.

3.2.1 Statistical methods

The *statistical methods* are based on patterns of word co-occurrence in texts of a sample collection (Jing & Croft, 1994). The methods assume that words that are contextually related, i.e., often appearing in the same sentence, paragraph, or document, are semantically related and hence should be classified in the same class. The more specific the context in which the words occur, the more precise the classification will be. A common procedure is to compute the similarity between a pair of terms based on coincidences of the terms in texts. When pair-wise similarities are available between all useful term pairs, an automatic term-classification process can collect all terms into common classes with sufficient large pair-wise similarities (Sparck Jones, 1971, p. 45 ff.). Among these term-classification strategies are single-link and complete link class-construction methods (Salton, 1989, p. 302). In a *single-link classification* system, each term must have a similarity exceeding a stated threshold value with at least one term in the same class. In the *complete link* or *clique classification,* each term has a similarity to all other terms in the same class that exceeds the threshold value. Alternatively, term classifications can be automatically constructed by adapting an existing document classification and by assuming those terms that occur jointly in the document classes could be used to form the desired term classes (cf. below learning of text classifiers). Peat and Willett (1991) argue against the utility of co-occurrence information in thesaurus construction. They observe that because synonyms often do not occur together in the same context, a co-occurrence based approach may have difficulty identifying synonymy relations. Although synonyms frequently do not co-occur, they tend to share neighbors that occur with both. Schütze and Pedersen (1994) define semantic closeness between terms as having the property of sharing common neighbors.

Statistically based thesaurus construction can yield acceptable results when learned from a large corpus of texts with a specialized vocabulary, but the technique is questionable with heterogeneous text databases (Salton & McGill, 1983, p, 228; Jing & Croft, 1994). Moreover, the technique simply detects associations between terms (e.g., synonyms and near synonyms, broader and narrower terms). Detecting the specific *nature of these associations* is usually beyond their scope.

3.2.2 Syntactic methods

The *syntactic methods* employ syntactic relations to determine semantic closeness of terms, A typical approach is to construct a hierarchical thesaurus from a list of complex noun phrases of a text corpus exploiting the head-modifier relationship of the noun phrases (Evans, Ginther-Webster, Hart, Lefferts, & Monarch, 1991). Here, the head is considered the more general term, which subsumes the more specific concept expressed by the phrase (e.g., "intelligence" subsumes "artificial intelligence"). Heads and modifiers are the smallest possible contexts of terms. Another example of constructing a thesaurus with syntactic information is to base a classification of nouns upon their being the subject of a certain class of verbs (Tokunaga, Iwayama & Tanaka, 1995). A better selection of terms that are syntactically associated can be obtained by combining the syntactic approach with statistical characteristics, such as the frequency of the associations (Ruge, 1991).

4. SUBJECT AND CLASSIFICATION CODES

4.1 Text Categorization

A *subject or classification code* is a general descriptor of the content of a class of texts. Its assignment is called categorization. Systems that automatically sort patterns into categories are called *pattern classifiers* or shortly *classifiers* (Nilsson, 1990, p. 2). The term *text classifier* is commonly used for a system that assigns subject and classification codes.

Humans perform a text categorization task by skimming the text and inferring the classes from specific *expressions* or word patterns and their *context* (see chapter 3). Automatic text categorization simulates this process and recognizes the classification patterns as a combination of text features. These patterns must be general enough to have a broad applicability, but specific enough to be consistently reliable over a large number of texts.

Automatic text categorization relates to a *knowledge-based approach*. A system that categorizes text needs a set of subject and classification codes and their relation with discriminating text features. It also needs matching or inference strategies to relate the surface features of a text to the category labels. The knowledge of the features and their corresponding class is manually acquired and implemented in a knowledge base, or is automatically learned from classified example texts.

4.2 Text Classifiers with Manually Implemented Classification Patterns

A knowledge base is an abstract representation of a topic area, or a particular environment, including the main concepts of interest in that area, and the various relationships between the entities. The construction of the knowledge base containing the patterns, concepts, and categorization rules is done by a *knowledge engineer* after careful analysis of texts in an example text base that is manually classified by experts (Sparck Jones, 1991). The classification patterns are thought to be predictable for new texts. A knowledge representation language or formalism is required that allows describing the domain of interest, expressing entities, properties, and relations (Edwards, 1991, p. 60 ff.).

1. The most common form for representing the text patterns and their relationships with the subject and classification concepts is by using *production or decision rules*. The condition usually involves single cue words, word stems, or phrases that logically combine using propositional or first-order logic. A rule has the form:
   IF <condition is true>
   THEN <assign category>
2. Occasionally, *frames* are used to represent the attributes of a particular object or concept in a more richly descriptive way than is possible using rules. The frame typically consists of a number of slots, each of which contains a value (or is left blank). The number and type of slots will be chosen according to the particular knowledge to be represented. A slot may contain a reference to another frame. Other features of frames have advantages: They include the provision of a default value for a particular slot in all frames of a certain type, and the use of more complex methods for "inheriting" values and properties between frames. When frames have mutual relationships, a *semantic net of frames* can represent them. Frames allow combining sets of related words with simple syntactic templates or with specifications that certain words occur within the same sentence, paragraph, or other context. They also allow representing semantic structures such as verbs describing classes of events.

The *actual classification* process simulates a *text skimming* for the cue patterns defined in the rule or frame base, possibly accompanied by an assessment of their attribute values and followed by an evaluation of the logical constraints imposed on them. The document text is only partially

parsed in order to detect the patterns, whereby the parsing is often restricted to a pattern matching procedure.

Knowledge bases have been proven *successful for classifying documents* in office environments (Chang & Leung, 1987; Eirund & Kreplin, 1988; Pozzi & Celentano, 1993; Hoch, 1994). But, also in broader subject domains such as categorization of news stories this approach proved to be fortunate (Mc Cune, Tong, Dean, & Shapiro, 1985; Young & Hayes, 1985; Riloff & Lehnert, 1994; Jacobs, 1993; Gilardoni, Prunotto, & Rocca, 1994). The famous CONSTRUE/TIS system (Hayes & Weinstein, 1991; Hayes, 1992) classifies a stream of Reuters economic and financial news stories into about 674 categories with precision and recall rates of the assignment of subject codes compared to expert assignments in the 90%.

Knowledge bases that describe the classification patterns and their relation with a subject or classification code have been successfully applied in text categorization. The results approximate human indexing, which proves that surface text features can be identified that successfully discriminate the subject and classification codes linked to a text. The knowledge representation is primarily controlled by semantic knowledge that often only characterizes a particular domain of discourse. When the number of patterns necessary to correctly categorize the texts of a document corpus is restricted, the construction and maintenance of a *handcrafted knowledge base* is a realistic task. In other circumstances, the machine learning methods discussed in the next sections provide an interesting alternative.

4.3 Text Classifiers that Learn Classification Patterns

Training a *text classifier* involves the construction of a classification procedure from a set of example texts for which the true classes are known. This form of learning is called *pattern recognition*, *discrimination*, or *supervised learning* (in order to distinguish it from *unsupervised learning* in which the classes are inferred from the data). The general approach is as follows. An expert, teacher, or supervisor assigns subject or classification codes to the example texts of the *training set*, which is also called the learning set or design set. It is assumed that this assignment is correct. Then, a classifier is constructed based on the training set. The aim is to detect general, but high-accuracy *classification patterns and rules* in the training set, which are highly predictable to correctly classify new, previously unseen texts. The set of new texts is called the *test set.* Because text classes are not mutually exclusive, it is convenient to learn a *binary classifier* for each class,

rather than to formulate the problem as a single multi-class learning problem.

More specifically, the archetypal supervised classification problem is described as follows (Bishop, 1995, p. 1 ff.; Hand, 1997, p. 5 ff.). Each object is defined in terms of a *vector of features* (often numerical, but also possible nominal such as color, presence or absence of a characteristic).

$$x=(x_1, x_2, \ldots, x_n) \tag{8}$$

where

x_j = the value that the feature j takes for object x

j = 1 .. n (n = number of features measured).

The features together span *a multi-variate space* termed the *measurement space* or *feature space.* For each object of the training set, we know both the feature vector and the true classes. The features of texts are commonly the words and phrases. The number of them is very large, creating the necessity of effective *feature selection* and *extraction* when training text classifiers. A text classifier learns from a set of positive examples of the text class (texts relevant for the class) and possibly from a set of negative examples of the class (texts non-relevant for the class). From the feature vectors of the examples, the classifier typically learns a classification function, a category weight vector, or a set of rules that correctly classifies the positive examples (and the negative examples) of the class. Each new text is equally represented as a feature vector, upon which the learned function, weight vector, or set of rules is applied to predict its class. Because, there are usually many classes and only few of them are assigned to a given example text, the number of negative examples in a training set exceeds the number of positive ones. Using negative relevance information is often a necessity when lacking positive relevance data.

Three broad groups of common training techniques can be distinguished for the pattern recognition problem (see Michie, Spiegelhalter, & Taylor, 1994): *statistical approaches*, *learning of rules* and *trees*, and *neural networks.* Another distinction can be made between parametric and non-parametric methods (Weiss & Kulikowski, 1991, p. 12 ff.). In the *parametric training methods,* the parameters are estimated from the training set by making an assumption about the mathematical functional form of the underlying population density distribution, such as a normal distribution. Then, the pattern discriminant functions that are used to classify new texts are based on these estimates. *Non-parametric training methods* make no such assumption about the underlying parameters. Here, the classification

functions initially have unspecified coefficients, which are adjusted or set in such way that the discriminant functions perform adequately on the training set. In work on learning in the artificial intelligence community, pattern recognition is often treated as one of *search (*Mitchell, 1977). The program is viewed as considering candidate functions or patterns from a search or hypothesis space, evaluating them in some fashion, and choosing one that meets a certain criterion. Searching a hypothesis space becomes especially explicit in methods that learn trees and rules.

Besides the many algorithms developed for pattern recognition, text categorization can draw upon more than 30 years experience of research into *relevance feedback* (Croft, 1995). In information retrieval, relevance feedback groups a number of techniques that learn a better query from the documents retrieved by the query and judged relevant or non-relevant for the query. Comparable to a relevance feedback strategy, in text categorization a concept is learned from the features of relevant and non-relevant texts.

The growing capacity of current machines and the increasing success of current learning algorithms enlarge the interest in the machine learning techniques (Feng & Michie, 1994; Croft, 1995). However, there are a number of challenges in applying traditional learning algorithms to text categorization, including a large and poorly defined feature set and an often low density of positive training examples (Hull et al., 1997).

In the next section we illustrate a selection of techniques that are most commonly used for training text classifiers.

5. LEARNING APPROACHES TO TEXT CATEGORIZATION

Because training text classifiers is such a broad topic, we treat it as a separate section in this chapter. We describe successively feature selection and extraction, statistical approaches to learning, learning of rules and trees, and training of neural networks. The division of training methods follows Michie, Tailor, and Spiegelhalter (1994). We illustrate the methods by a description of the most common algorithms. Some general results of applying the classification algorithms upon new texts are given in terms of recall, precision, and F-measures. The results are sometimes hard to compare across difficult text collections, which are possibly preprocessed by different feature selection techniques and have various sizes of training sets.

5.1 Feature Selection and Extraction

5.1.1 Feature selection

Feature selection aims at eliminating low quality features and at producing a lower dimensional feature space. The real need for feature selection arises for problems with a large number of features and with relatively few samples of each class to be learned (Weiss & Kulikowski, 1991, p. 72 ff.), which is the case in text categorization. Feature selection is done manually by human experts or with automated tools, the latter usually being applied in text categorization.

A limited number of features is advantageous in classification (Weiss & Kulikowski, 1991, p. 72 ff.). A limited feature set benefits efficiency and decreases computational complexity. It reduces the number of observations to be recorded and the number of hypotheses to test in order to find an accurate classifier. But more importantly, a small feature set decreases the danger of *overfitting* or overtraining. Overfitting means that the learned classifier perfectly fits the training set, but does not perform well, when applied upon new, previously unseen cases. The classifier fails to generalize sufficiently from the training data and is too specific to classify the new cases. A large number of features enhances this effect. So, when using many features we need a corresponding increase in the number of samples to ensure a correct mapping between the features and the classes. The property of too many features is known as the *dimensionality problem* (Bishop, 1995, p. 7 ff.; Hand D.J., 1997, p. 3 ff.).

Feature selection removes redundant and noisy features. Noise is defined as erroneous features in the description of the example (Quinlan, 1986) or as features that are no more predictive as by chance (Weiss & Kulikowski, 1991, p. 11). Noisy features lead to overfitting and to poor accuracy of the classifier to new instances.

Feature selection is done before training, during training, and after classification of new, previously unseen objects. When done *before training*, it is usually the quality of an individual feature that is evaluated and the feature is removed from the feature set after a negative evaluation. *During training*, some algorithms incorporate a feature selection process. This is especially true for algorithms that induce decision trees or rules from the sample data. They often include stepwise procedures, which incrementally add features, discard features, or both, evaluating the subset of features that would be produced by each change. Feature selection is done *after classification of new objects* by measuring the error rate of this

classification. Those features are removed from or added to the feature set when this results in a lower error rate on the test set. The choice of a feature selection technique is usually application-specific and domain knowledge is considered important in the feature selection process (Nilsson, 1990, p. 4).

5.1.2 Feature extraction

Feature extraction, also called *re-parameterization*, creates new features by applying a set of operators upon the current features (Hand D.J., 1997, p. 15 1 ff.). Although a single feature can be replaced by a new feature, it more often occurs that a set of features is replaced by one feature or another set of features. Logical operators such as conjunction and disjunction can be used. Operators such as the arithmetic mean, multiplication, linear combination, and threshold functions can be sensibly applied to many numeric functions. When a set of original features are thought to be redundant manifestations of the same underlying feature, replacing them with a single feature corresponding to their sum, disjunction, mean, or some other cumulative operation is a good approach. Often, operators that produce a linear transformation of the original features are used (e.g., factor analysis). Operators can be specific to a particular application and domain-knowledge is considered important in a feature extraction process (Bishop, 1995, p. 6). Feature extraction can be done *before training,* when the original features of each example object are transformed into more appropriate features. Feature extraction can also be *part of training,* such as the computation of a feature vector for each class from the feature values of the individual examples.

5.1.3 Feature selection in text categorization

The salient features of a text in a classification task are its words and phrases. The training of a text classifier is often preceded by an initial *elimination of a number of irrelevant features.* Individual features are judged and possibly removed. A feature can be eliminated with respect to its overall relevance in determining a text's content, or with respect to its value in determining a particular category.

As it is seen in chapter 4, the words and phrases of a text do not contribute equally to its content. So, similarly to the process of *extracting content terms from texts,* a number of text features can be selected from a text that are supposed to reflect its content. The techniques include the elimination of stopwords and weighting words and phrases according to their distribution characteristics, such as frequency of occurrence (e.g., Cohen, 1995), position on a Zipf curve (e.g., Sahami, Hearst, & Saund, 1996), fit to

a Poisson distribution (e.g., Ng, Loewenstern, Basu, Hirsh, & Kantor, 1997), followed by a removal of words with a low weight. Aggressive removal of words with a domain-specific stopword list is sometimes used (e.g., Yang, 1995; Yang & Wilbur, 1996). The opposite is using a list of valid feature terms from a domain-specific dictionary. Knowledge of the discourse structure and of the value of certain text positions or passages for feature selection is considered important, especially in long texts (e.g., Maron, 1961; Borko & Bernick, 1963; Fuhr, 1989; Jacobs, 1993; Apté, Damereau, & Weiss, 1994; Yang, Chute, Atkin, & Anda, 1995; Thompson, Turtle, Yang, & Flood, 1995; Brüninghaus & Ashley, 1997; Leung & Kan, 1997).

Feature selection with respect to the relevance in determining a text's content involves a reduction of the dimensionality of the global feature space. Such an initial selection is often not sufficient as feature selection technique given the large number of text features. So, there are a number of useful techniques that select the features per class to be learned and that take into account the *distribution of a feature in example texts that are relevant or non-relevant for the subject or classification code*. In general, a good feature is a single word or phrase that has a statistical relationship with a class, i.e., having a high proportion of occurrences within that particular class and a low proportion of occurrences in the other classes. Many feature selection techniques for training text classifiers are based upon this assumption. The techniques compute for each class the *relevance score* of a text feature, i.e., strength of the association between the class concept and the feature, and eliminate features with a low score. Many of the scoring functions originated from relevance feedback research, but have been used in text categorization (e.g., Maron, 1961; Field, 1975; Hamill & Zamora, 1980; Voorhees & Harman, 1997). A feature typically is ranked by the difference in relative occurrence in relevant and non-relevant texts for the subject or classification code (Allan et al., 1997) and by its difference in mean weights in relevant and non-relevant texts for the subject or classification code (Brookes, 1968; Robertson, Walker, Beaulieu, Gatford, & Payne, 1996; cf. Rocchio algorithm below). Finally, there are the techniques that assume a probability distribution of the feature in the example set and employ the deviations from this distribution in feature selection. Leung and Kan (1997) use the deviation of the value of the feature from its mean value in the example set normalized by the standard deviation (*z-score*). The χ^2 (*chi-square*) test measures the fit between the observed frequencies of the features in texts of the example set and their expected frequencies (i.e., the terms occur with equal frequencies in texts that are relevant for the class and in texts that are non-relevant for the class) and identifies terms that are strongly related to the text class (Cooper, Chen, & Gey, 1995; Schütze, Pedersen, & Hearst, 1995; Schütze,

Hull, & Pedersen, 1995; Hull et al., 1997). In still another technique, a *binomial probability distribution* is used to compute the probability that a text feature occurs in the texts relevant for the subject or classification code purely by change and to relate a low probability with a high descriptive power for the text class (Yochum, 1995; cf. Dunning, 1993).

Although feature selection is an absolute necessity in text categorization, caution must be taken in removing text features. For some text classes, words that seem to have low overall content bearing value can be an important category indicator especially in combination with other terms (Riloff, 1995; Jacobs, 1993; cf. Hand D.J., 1997, p. 150).

5.1.4 Feature extraction in text categorization

There are a number of feature extraction techniques that can be employed in text categorization. The process of *stemming* (see chapter 4) reduces a number of text features to one single term or feature (e.g., Schütze, Hull, & Pedersen, 1995). The *phrase formation* process is sometimes seen as feature extraction. A phrase groups original single words of a text that are statistically and/or syntactically related (e.g. , Finch, 1995). *Weighting* of the original text feature aims at increasing the predictive value of the feature. Weighting includes the traditional weighting schemes for content identification (e.g., term frequency (*tf*), inverse document frequency (*idf*), and *tf* x *idf*) (e.g., Yang & Chute, 1994) and the relevance scoring functions that determine the weight of a term for a text class (see above). *The use of thesauri* also transforms the original text features with more uniform and more general concepts (e.g., Blosseville, Hébrail, Monteil, & Penot, 1992), whereby the groups of semantically related words can automatically be built (e.g., Baker & McCallum, 1998). *Latent Semantic Indexing* (*LSI*) replaces the text features (usually words) of a document set by their lower dimensional linear combination. This is done by *singular value decomposition* of the feature by document matrix (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990; use of LSI in text categorization: Hull, 1994; Dumais, 1995; Schütze, Hull, & Pedersen, 1995).

5.1.5 A note about cross validation

Another technique to overcome the *overfitting* problem is *cross validation* (Henery, 1994), wherein the parameters of the model are updated based on the error detected with the validation set. A part of the training set (e.g., two thirds) is used for training the classifier, while the remainder is used as the validation set. During training, errors in classifying the validation

set help in selecting features or in determining when overfitting has occurred. The latter refers to training procedures, which iterate to find a good classification rule (e.g., training of a neural network). At each iteration, the parameters of the model are updated and the error is computed upon the validation set. Training continues until this error increases, which indicates that overfitting has set in (cf. Schütze, Hull, & Pedersen, 1995).

5.2 Training with Statistical Methods

We focus upon discrimination techniques, *k*-nearest neighbor classifiers, and Bayesian independence classifiers. *Discrimination techniques* aim at finding a function that in a best way separates the examples of two classes. Many of the newer discrimination techniques attempt to provide an estimate of the joint distribution of the features within each class, which can in turn provide a classification rule. *k-nearest neighbor classifiers* do not generalize the examples into an explicit rule or function, but use specific examples to generate classification predictions. *Bayesian independence classifiers* estimate the posterior probability that a document text belongs to a specific class given its features. The advantage of statistical approaches in text categorization is that they provide a probability of being in each class, rather than simply a classification.

5.2.1 Discrimination techniques

Discriminant analysis aims at determining a function that discriminates between classes (Michie et al., 1994, p. 17 ff.; Hand D.J., 1997, p. 23 ff.). *Linear discriminant analysis* finds a linear combination of the features (variables) of the example objects that separates two classes in a best possible way. A hyperplane (line in two dimensions, plane in three dimensions, etc.) in the *n*-dimensional feature space is chosen to separate the classes. In text categorization, for each class a function is sought that separates the example texts relevant for the class from those example texts non-relevant for the class. A common discrimination technique in training text classifiers is the *linear discriminant by least squares (*e.g., Schütze, Hull, & Pedersen, 1995; Lai, Lee, & Chew, 1996; Blosseville et al., 1992). Here, the hyperplane is sought for which the sums of squares deviated of the feature values is minimal. The feature vector of a new object to be classified represents a specific point in the *n*-dimensional vector space. According to the side of the hyperplane it falls on, the appropriate class is assigned. Logistic *discriminant analysis* or *logistic regression* is another discriminant analysis technique employed in training text classifiers (e.g., Gey, 1994;

Fuhr & Pfeifer, 1994). Logistic regression usually starts with the linear discriminant function and then iteratively adjusts the function in order to find a logistic function that better fits the data and better separates the classes. With a large feature set, discriminant analysis imposes severe computational and storage complexity. Evaluation of the categorization yields recall and precision values of about 50% or less compared to expert assignment, when training is based upon a heterogeneous text corpus (Schütze, Hull, & Pedersen, 1995; Lai et al., 1996). Results improve (up to about 80% in precision) when training is based upon texts of a limited subject domain (Blosseville et al., 1992).

Instead of constructing a hyperplane that separates the classes, *alternative discrimination techniques* provide a classifying rule for each class. The rule is based on the joint distribution of features in the positive example set and is possibly based on the joint distribution of features in the negative example set. The methods combine feature extraction with training. The result of the training is often a *weight vector or solution vector for each class*. The weight vector is built based on the individual feature vectors of the example objects. Each component of the vector represents a text feature and its weight regarding the class. The weight vector classifies all of the examples correctly and is situated somewhere, preferably in the middle, of the solution region of the class that is learned in the *n*-dimensional solution space (Duda & Hart, 1973, p. 138 ff.).

The construction of category weight vectors is common in *text categorization.* When a new text is classified, a *scoring function* is computed taking into account the feature vector of the new text and the weight vector of each class. When the resulting *score* for a specific category weight vector exceeds some threshold or satisfies some other criterion, the category is assigned to the new text. The scoring function is usually a *distance or similarity function* (see Jones & Furnas, 1987) applied upon the two vectors. The resulting ranking yields a probabilistic membership of the new text for the class and allows detecting when no category weight vector is closely enough related. The simplest scoring functions are *linear*, i.e., they are expressed as the *inner or dot product* of a weight vector w and a feature vector x (Lewis, Schapire, Callan, & Papka, 1996):

$$f(x) = w \cdot x = \sum_{j=1}^{n} w_j x_j \tag{9}$$

where

j = text feature

n = number of features measured.

Linear classifiers are simple to construct and use. They produce a classification by comparing a weighted sum (a linear combination) of the measurements with a threshold. The weights in the sum represent how important each variable is in determining the overall score, when taken in conjunction with the other variables in the sum. Sometimes a *non-linear ranking* is proposed such as the *cosine function* (e.g., Masand, Linoff, & Waltz, 1992).

The following illustrates the construction of a category weight vector with two common algorithms.

5.2.2 An illustration: The Rocchio algorithm

The *Rocchio algorithm* is originally developed for relevance feedback in information retrieval to improve existing queries (Rocchio, 1971). But, it is used in a like manner to classify texts (Buckley, Salton, Allan, & Singhal, 1995; Larkey & Croft, 1996; Lewis et al., 1996; Brüninghaus & Ashley, 1997). For each category, a weight vector is computed. The weight of a feature j (w_j) in the weight vector is computed as the weighted difference of the mean belief or importance of the feature in positive and negative training examples of the class C_k:

$$\beta \cdot \frac{1}{n_{relC_k}} \sum_{relC_k} belief - \gamma \cdot \frac{1}{n_{nonrelC_k}} \sum_{nonrelC_k} belief \qquad (10)$$

where

belief =	the importance assigned to a text feature in an example text (eg, term weight)
$nrelC_k$ =	the number of example texts relevant for class C_k
$nnonrelC_k$ =	the number of example texts non-relevant for class C_k
β =	the importance assigned to the mean belief of the feature j in the example texts relevant for class C_k
γ =	the importance assigned to the mean belief of the feature j in the example texts non-relevant for class C_k where usually $\gamma < \beta$.

In text categorization the Rocchio algorithm has moderate results. For instance, the experiments of Lewis et al. (1996), in which system assignment of classification codes to medical texts and news stories is compared with expert assignment, resulted in F-measures ($\beta = 1$) lower than 50%. But, the Rocchio algorithm is found to be a good choice as training algorithm with classes for which the number of positive examples is low.

5.2.3 An illustration: The Widrow-Hoff algorithm

To learn a category weight vector, the *Widrow-Hoff algorithm* (Duda & Hart, 1973, p. 156; Hertz, Krogh, & Palmer, 1991, p. 103 ff.; Lewis et al., 1996) runs through the example objects one at the time updating a weight vector at each step. Initially the weight vector is a vector chosen at random. At each step, a new weight vector w_{i+1} is computed from the old weight vector w_i by using training example x_i. The j^{th} component (feature j) of the new weight vector w_{i+1} for class C_k is found by applying the rule:

$$w_{i+1,\,j} = w_{i,j} - 2\eta(w_i.x_i - y_i)x_{i,j} \qquad (11)$$

where

w_i = the old weight vector

w_{ij} = the value of the j^{th} component (feature j) of vector w_i

η = learning rate, which controls how quickly the weight vector w will change, and how much influence each new example has on it ($\eta > 0$)

x_i = the feature vector of the presented training example

y_i = a value indicating whether or not the class C_k that is learned applies to the training example x_i (usually 1 or 0)

x_{ij} = the value of the j^{th} component (feature j) of vector x_i.

This algorithm is usually viewed as a *gradient descent procedure* (Bishop, 1995, p. 263 ff.). At each iteration, the new weight vector w_{i+1} is moved some distance from the old weight vector w_i in order to minimize the square or error $(w_{i+1}.x_i - y_i)^2$ of the new weight vector on the current example x_i. To accomplish this effect, the rule will consider the error $(w_i.x_i - y_i)^2$ of the old weight vector w_i on the current example x_i and adapt the old weight vector accordingly. The weight update rule minimizes the error by using its *gradient*. The term $2(w.x - y)\,x$ is the gradient (with respect to w) of the square loss $(w.x-y)^2$. So, the old weight vector is moved in the direction of the steepest descent, i.e., along the negative of the gradient, which is the direction in which the error is (locally) decreasing the fastest. The *gradient descent procedure* is a procedure employed for the *backpropagation algorithm* in training a neural network.

This algorithm has been employed for training text classifiers. It was trained upon medical texts (Medline records) and newswire stories, and applied upon new, previously unseen texts. The results were compared with the category assignments by experts (Lewis et al., 1996). The F-measure ($\beta = 1$) varies from 1% to 72% depending upon the text corpus, the feature selection and extraction technique employed, and the number of positive

training instances, but a better performance is obtained than with the Rocchio training algorithm in the same circumstances.

5.2.4 *k*-nearest neighbor classifiers

Another important statistical technique in training classifiers is the *Nearest Neighbor* (*NN*) approach (Duda & Hart, 1973, p. 103 ff.; Weiss & Kulikowski, 1991, p. 70 ff.), which is also called *Memory Based Reasoning* (Stanfill & Waltz, 1986; Masand et al., 1992) or *Instance-Based Learning* (*IBL*) (Aha, Kibler, & Albert, 1991).

A *k-nearest neighbor classifier* will not learn by generalizing examples into an explicit abstraction such as a function or rule that separates the positive and negative examples of a class. It only stores the original positive examples of a class. When a new case arrives, the nearest neighbor classifier compares the feature vector of the new case with the feature vector of each example stored. The classifier assumes that similar instances have similar classifications. The classifier finds the closest examples with which the similarity exceeds a certain threshold and pick up the class of these for the new case. Alternatively, it will find the *k* (some constant) closest examples. The inner product and the cosine function are commonly used for the vector comparisons. Geometrically, there is no general form to draw a boundary between the classes, because the nearest neighbor method can produce any arbitrarily complex surface to separate the classes based only on the configuration of the sample points and their similarity or distance metric to one another.

Nearest neighbor classifiers have *advantages* (Stanfill & Waltz, 1986).

1. Provided a good example, the classifier can form a hypothesis for a new case on this single precedent.
2. The lack of generalization of examples obviates the need to store rigid generalizations in category concept descriptions, so a more flexible matching between the new case and the training set is possible.
3. Nearest neighbor classifiers can learn multiple, possibly overlapping classes simultaneously.

Nearest neighbor classifiers have *disadvantages* (Stanfill & Waltz, 1986).

1. The nearest neighbor method involves almost no effort in learning from the training set. But, the classification time of a new case is large: The new case must be compared with each example of the training set,

sometimes resulting in a need for parallel execution of the comparisons (Masand et al., 1992).

2. The classifier also requires large storage for all the examples.
3. Nearest neighbor classifiers demand an accurate feature set. They generally perform well with good predictive features, but are intolerant of irrelevant or noisy features. A number of techniques have been proposed to relieve the impact of noisy features. Instances may be averaged (cf. the construction of a category weight vector) (Kibler & Aha, 1990; Dumais, 1995; Oard, 1997). Or, only good examples that proved to perform well during classification of new cases are stored for further comparisons (Aha et al., 1991). Another solution is to provide a large number of examples for each category to be learned in order to account for the noise in the texts (Creecy et al., 1992).

The results of applying the *k*-nearest neighbor classifier in text classification are good when the number of training examples is very large (e.g., Masand et al., 1992). Classification of news stories yields recall and precision values in the range of 70-80%, when compared with an expert classification. In other circumstances, recall and precision values remain below 50% (e.g., Yang, 1994; Larkey & Croft, 1996).

5.2.5 Bayesian independence classifiers

The general model of the *Bayesian independence classifier* can be described as follows. A small set of features is selected for each class. The posterior probability that a new, previously unseen case belongs to a certain class given the features of the case is computed based on the probabilities that these individual features are related to the class. Probability estimates of the individual features are based on the co-occurrence of classes and the selected features in the training corpus, and on the assumption of their linkage. The computation of the probability that the new case belongs to a specific class is simplified by using the *theorem of Bayes,* which assumes that the probabilities of the features are independent. Class membership is assigned to the new case when the probability of class membership is higher than a pre-set threshold or when the class belongs to the top *k* (some constant) classes proposed. Sometimes a proportional factor (a priori class probability) is used in the computation: A class is assigned in proportion to the number of times it is assigned in the training set.

The Bayesian independence classifier was first proposed by Maron (1961) as a way to estimate the probability that a subject or classification code should be assigned to a document text given the presence of cue words

in the text. Various improvements to Maron's approach have been explored (e.g., Fuhr, 1989; Fuhr & Buckley, 1991; Lewis, 1992a, 1992b, 1995; Del Favero & Fung, 1994; Lewis & Gale, 1994). Bayes' theorem for assignment of the class C_k given the conditioning event x is:

$$P\left(C_k = 1 \middle| x\right) = \frac{P\left(x \middle| C_k = 1\right) \cdot P\left(C_k = 1\right)}{P\left(x\right)} \tag{12}$$

where

$P(C_k=1)$ = a priori class probability of category C_k being assigned in the training set.

$$P\left(C_k = 1 \middle| w_1 = x_1, \ldots, w_p = x_p\right) = \frac{P\left(w_1 = x_1, \ldots, w_p = x_p \middle| C_k = 1\right) \cdot P\left(C_k = 1\right)}{P\left(w_1 = x_1, \ldots, w_p = x_p\right)} \tag{13}$$

where

$w_1, \ldots, w_p$ = set of p terms chosen as predictor features.

The model of Maron (1961) considers only the presence ($x_i = 1$) of a term. The probability that the class C_k is assigned given document text D_m is computed as :

$$P\left(C_k = 1 \middle| D_m\right) = \frac{P\left(w_1 = 1, \ldots, w_p = 1 \middle| C_k = 1\right) \cdot P\left(C_k = 1\right)}{P\left(w_1 = 1, \ldots, w_p = 1\right)} \tag{14}$$

Assuming independence of features in (14) yields :

$$P\left(C_k = 1 \middle| D_m\right) = P\left(C_k = 1\right) \cdot \prod_j \frac{P\left(w_j = 1 \middle| C_k = 1\right)}{P\left(w_j = 1\right)} \tag{15}$$

where

$P(w_j=1|C_k=1)$ = the probability that the feature w_j is present in a text of the example set that is relevant for the class C_k

$P(w_j=1)$ = the probability that the feature w_j occurs in the complete training set.

The model of Fuhr (1989) and Lewis (1992a, 1992b, and 1995) considers the presence ($x_i = 1$) and the absence of a term ($x_i = 0$). The probability that the class C_k is assigned given document text D_m is computed as :

$$P(C_k = 1|D_m) = \sum_{x \in \{0,1\}} \frac{P(w_1 = x, \ldots, w_p = x | C_k = 1) \cdot P(C_k = 1)}{P(w_1 = x, \ldots, w_p = x)} P(x|D_m) \tag{16}$$

Assuming independence of features in (16) yields :

$$P(C_k = 1|D_m) =$$

$$P(C_k = 1) \cdot \prod_j \left(\frac{P(w_j = 1|C_k = 1) \cdot P(w_j = 1|D_m)}{P(w_j = 1)} + \frac{P(w_j = 0|C_k = 1) \cdot P(w_j = 0|D_m)}{P(w_j = 0)} \right) \tag{17}$$

where (see also (15))

$P(w_j = 0|C_k = 1) =$ the probability that the feature w_j is not present in a text of the example set that is relevant for the class C_k

$P(w_j = 0) =$ the probability that the feature w_j does not occur in the complete training set

$P(w_j = 1|D_m) =$ the probability that the feature w_j is present in D_m

$P(w_j = 0|D_m) =$ the probability that the feature w_j is not present in D_m.

Conditional independence is often not a valid assumption for text features. When words of a text occur in the same sentence or paragraph, this independence is sometimes difficult to hold. The Bayesian independence classifier is useful when a limited feature set is identified. The independence assumptions are increasingly violated as more features are used, An alternative for employing this probabilistic model requires complete probability data for all statistical dependencies among all text features, which for many, especially heterogeneous text corpora is impossible to compute.[1] The results of applying the Bayesian independence classifier show recall and precision values of about 50% compared to expert assignments (e.g., Larkey & Croft, 1996).

Other interesting statistical techniques for text classification regard *linear regression methods* (Borko & Bernick, 1963; Yang & Chute, 1993, 1994).

5.3 Learning of Rules and Trees

Learning of rules and trees aims at inducing classifying expressions in the form of decision rules and trees from example cases. The rules and trees are capable of categorizing new, unseen cases (Weiss & Kulikowski, 1991; Quinlan, 1993; Feng & Michie, 1994). Each decision rule is associated with a particular class, and a rule that is satisfied, i.e., evaluated as true, is an indication of its class. Thus, classifying new cases involves the application of the learned classifying expressions and assignment to the corresponding class upon positive evaluation.

The training examples are represented as a *set of features* or as a *set of relations between features.* The learned classifying expressions can take the form of decision rules or trees. Rules are of the form if-then. They are expressions in *propositional logic* or in *first-order logic* that can be evaluated as true or false. A decision tree partitions samples into a set of covering decision rules. A decision tree consists of nodes and branches. Each node, except for terminal nodes or leaves, represents a test or decision and branches into subtrees for each possible outcome of the test. The tree can be used to classify an object by starting at the root of the tree and moving through it until a leaf (class of the object) is encountered. When the decision rules are not mutually exclusive, the decision or production rule format leads to a more efficient and compact coverage of the classes (Weiss & Kulikowski, 1991, p. 133). This explains the preference to induce decision rules instead of trees in text categorization.[2]

The general process of *rule induction is as follows.* The rules are found by searching these combinations of features or of feature relations that are discriminative for each class. Given a set of positive examples and a set of negative examples (if available) of a class, the training algorithms generate a rule that covers all (or most) of the positive examples and none (or fewest) of the negative examples. Having found this rule, it is added to the rule set, and the cases that satisfy the rule are removed from further consideration. The process is repeated until no more example cases remain to be covered.

The are two major ways for accessing the *search space* of features and a third combined way (Mitchell, 1977; Feng & Michie, 1994). *General-to-specific* methods search the space from the most general towards the most specific hypothesis. One starts from the most general rule possible (often an empty clause), which is specialized at the encounter of a negative example that is covered. The principle is of adding attributes to the rule. An example is the FOIL algorithm (Quinlan, 1990). *Specific-to-general* methods search the hypothesis space from the most specific towards the most general hypothesis and will progressively generalize examples. One starts with a

positive example, which forms the initial rule for the definition of the concept to be learned. This rule is generalized at the encounter of another positive example that is not covered. The principle is of dropping attributes. An example is the GOLEM algorithm (Muggleton & Feng, 1990 cited in Feng & Michie, 1994). The combination of the general-to-specific and the specific-to-general methods is the so-called *version space method*, which starts from two hypotheses (Mitchell, 1977). Negative examples specify the most general hypothesis. Positive examples generalize the most specific hypothesis. The learning process stops when both hypotheses converge to one concept description. The hypothesis is *complete* when it covers all positive examples, and it is *consistent* when it does not cover any negative ones. It is possible that a hypothesis does not converge to a (nearly) complete and (nearly) consistent one, indicating that there is no rule that discriminates between the positive and the negative examples. This can occur either for noisy data, or in case where the rule language is not sufficiently complex to represent the dichotomy between positive and negative examples.

The version space model suffers from practical and computational limitations. To test all possible hypotheses is often impossible given the number of feature combinations. The research focuses on how to reduce the search space while still obtaining a complete and consistent hypothesis.

1. By searching a rule that covers most of the positive examples and removal of the examples from further training, the search space is divided into subspaces, for each of which a covering rule is sought.
2. Simple rules are often preferred above complex ones.
3. The search space is often restricted by considering a single best feature for inclusion or exclusion at each stage of building a rule. Because backtracking is not used, i.e., each new choice depends on the previous choices, a good but not always optimal set of classifying rules is obtained. Such non-backtracking algorithms are called *greedy algorithms* (Quinlan, 1993, p. 20).
4. When full backtracking is used, it is possible to organize the search so that relatively few possibilities must be examined. The *branch and bound* method will not consider a set of hypotheses if there is some criterion that allows assuming that they are inferior to the current best hypothesis.

Notwithstanding its computational complexity, learning of rules and trees has *advantages* that are highly valued in text categorization.

1. Modeling a classifier in a form that is compatible with human-expressed knowledge is beneficial. Human-engineered rule-based systems are

successful in categorizing document texts (see above). If accurate rules can be learned from example texts, we can expect similar results when the rules are automatically learned. Additionally, induced production rules can easily be supplemented with handcrafted knowledge (e.g., common knowledge) or be verified.

2. An important circumstance that might favor applicability of rule and tree induction methods is the presence of conditional dependencies among features (Feng & Michie, 1994). Such dependencies are often present in texts (e.g., dependencies between individual words of a text, especially between the words in a sentence or paragraph).
3. But, the technique requires a limited and accurate feature set in order to reduce the computational complexity and to obtain a good hypothesis when not all hypotheses are tested.

Induction of rules is promising in text categorization. An example of a *propositional learner* is discussed in Apté et al. (1994) (see also Weiss & Indurkhya, 1993 for details on the training algorithm) and yields excellent precision and recall values of about 70-85% compared to expert assignments when applied upon Reuters newswires. Cohen (1995) learns a *first-order logic* text classifier with the algorithm FOIL6 (Quinlan, 1990). First order logic allows formulating rules that incorporate relations between text features (e.g., relation of nearness or succession between the words of the text). When applied upon the short texts of newswire headlines, the classifier yields an average recall of about 47-50%, an average precision of about 35-55%, and an average F-measure ($\beta = 1$) of about 36-50% depending on the feature selection technique used. These results are slightly better than propositional learning with FOIL under the same circumstances.

5.4 Training with Neural Networks

A *neural network* (*NN*) (Hertz, Krogh, & Palmer, 1991) is a network of (neuron-like) units or nodes, some of which are designated as input or output units. The units have weighted connections and units may have a bias. A simple model works at follows. To process a case, the input units are first assigned feature values. At a given input the activation will spread over the network. Each unit computes a weighted sum of its input data augmented by the unit's bias. It outputs this value to its further connections when this value satisfies a certain criterion (e.g., threshold) or after this value is subjected to a general non-linear function, called *gain* or *activation function.* Values of the output units determine the classes to which the case belongs. The high connectivity of the network (i.e., the fact that there are many terms in the

sum) means that errors in a few terms will probably be inconsequential. So, a neural network is expected to show a certain robustness in the presence of noise and errors.

Neural networks are popular because of their ability to *generalize* to new situations. They can be trained on a number of example cases. Network weights and biases are learned through repeated examination of example cases. Neural networks are trained by *backpropagation.* The activation of each input pattern is propagated forward through the network, and the error produced is then backpropagated and the parameters changed so as to reduce the error. More specifically, the deviation of each unit's output from its correct value for the case is backpropagated through the network; all relevant connection weights and unit biases are adjusted to make the actual output closer to the target. One of the simplest algorithms for these adjustments is the *gradient descent rule* (cf. (11)) (Bishop, 1995, p. 253 ff.). Training continues until the weights and biases stabilize. Neural networks have the ability to fit a range of distributions accurately.

Neural nets are *not very commonly used for training text classifiers.* Schütze, Hull, and Pedersen (1995) use a neural network trained with the gradient descent rule for a text routing problem (e.g., routing of newswire, patents, and scientific abstracts). Average precision of routing new texts (about 40-50% compared to expert routing) is higher than when training the classifier with linear discriminant analysis or logistic regression techniques. These experiments also demonstrate the need for good feature selection and extraction when training neural nets for text categorization in order to reduce the computational complexity.

6. ASSIGNMENT OF CONTROLLED LANGUAGE INDEX TERMS: ACCOMPLISHMENTS AND PROBLEMS

The assignment of thesaurus class terms and of subject and classification codes to texts follows a general strategy. The controlled language index terms are inferred from the actual words and phrases of the text. Assignment of the terms requires knowledge of the relationships between the concepts of the text and the text features. Manual construction and maintenance of knowledge sources, i.e., of thesauri and knowledge bases, are expensive tasks. Moreover, given the rate at which texts are currently produced, there is a need to easily adapt the knowledge to changing document collections and classification systems. Automating the knowledge acquisition allows

adapting to these changes more easily and detecting patterns that are sometimes not obvious to a knowledge engineer.

Automated knowledge acquisition has its own *problems*. Constructing thesauri automatically is very difficult. It is especially hard to define automatically the kind of relationship that holds between terms. Training a text classifier upon example texts is promising, but is not without difficulties. Firstly, there is the problem of the *large number of features* when recognizing patterns in texts. These features are often *noisy*, i.e., irrelevant to the patterns to be learned. Among the noisy features, there are the words and phrases of a text that are of no or little relevance in identifying the topics. Additionally, texts often discuss different topics. Secondly, trainable text classifiers are often confronted with *few positive examples* of patterns to be learned. It is acknowledged that feature selection and extraction relying upon prior knowledge about the texts is important in text classification.

7. CONCLUSIONS

Controlled language index terms are valuable index terms. When automating their assignment to texts, the knowledge about the words and phrases that imply the term concepts is needed. This knowledge is implemented in thesauri and knowledge bases for text categorization. Building thesauri automatically remains a very difficult task. Learning the classification patterns of broad text classes is somewhat easier. Constructing a text classifier that generalizes from example texts can build upon a long tradition of research in pattern recognition and of experiments in relevance feedback in retrieval. The problem is to correctly find the patterns in example texts that are associated with the subject or classification codes. Statistical techniques of pattern recognition, leaning of rules and trees, and training of neural nets are all based upon the principle that when a large number of examples or a limited number of good instances are available, the desired patterns will be identified based upon re-occurring features, and noise will be neglected. However, in text classification the number of features is enormous and many features have no relevance. In addition, the number of positive examples of each text class is often limited due to changing document collections and classification systems.

---

[1] Van Rijsbergen (1977) gives a theoretical model for estimating dependencies between words from the distribution of occurrences and co-occurrences in a corpus of example texts.

[2] Examples of the use of classification trees employed for routing documents are: Crawford, Fung, Appelbaum, and Tong, 1991; Tong, Winkler, and Gage, 1993.

Chapter 6

AUTOMATIC ABSTRACTING: THE CREATION OF TEXT SUMMARIES

1. INTRODUCTION

Abstracting generates a summary of a text's content, which has various possible formats. The main formats are a short coherent text and a text profile. Both reduce the content of the source text to its essentials, but the profile structures the important content of the text in semantically well defined fields. An abstract represents the text in a more comprehensible way than index terms, making the abstract especially suited to document and information selection tasks.

The idea of *automating text summarization* again goes back to Luhn (1958). Besides some limited efforts in the 1960s, text summarization has never received special attention, apart from the application of artificial intelligence techniques in restricted subject domains. However, with the current information overload, the topic has received renewed interest. This interest is fed by improvements of natural language processing methods that extend to whole texts.

Automatic abstracting of text consists of three steps (Figure 1). The *text analysis step* identifies the essential content of the source text resulting in a source text representation. In the *transformation step* the content of the source text is condensed either by *selection* or *generalization* of what is important in the source. The selected and generalized information is captured in a summary representation. Finally, the *synthesis step* involves *drafting and generation of the summary text* based upon the summary representation.

This last step is especially concerned with the organization of the content and the presentation of the abstract. Function and audience of the summary determine relevancy in the source text and the format of the output.

In this chapter the techniques for text abstracting are discussed in relation to these three steps. The main focus is upon analysis of and information selection from the source text. These steps are important for generating adequate text representations. Research efforts both encompass implementing symbolic, knowledge-based techniques as well as shallow statistical approaches that rely upon word distributions or statistical methods for learning discourse patterns. It seems easy to build a text summarization system for one particular application. On the other hand, it seems impossible to have a single general technique for automatic summarizing given the variety of input texts and possible summary descriptions. But, the real challenge is to develop *techniques*, which are *general* in being *relevant to well-defined classes of situations*. This chapter starts with a short description of the evaluation of automatic abstracting. Again, we finish the chapter by enumerating the accomplishments and problems of the techniques.

2. A NOTE ABOUT EVALUATION

Automatic text summarization is usually seen as a natural language processing task. Evaluation is more complex than evaluation of index terms. The criteria applied in performance evaluation of abstracts fall under two major heads, intrinsic and extrinsic. Intrinsic criteria are those relating to a system's objective, extrinsic criteria are those bearing upon its function, i.e., its role in relation to its setup's purpose. Much more than indexing systems, abstracting systems rely upon knowledge sources (linguistic, domain, and contextual), so criteria that judge the performance of knowledge base software and text parsing become important.

An *intrinsic evaluation* (Sparck Jones & Galliers, 1996, p. 19 ff.) judges the quality of the abstracts directly based on judgements of informativeness, coverage, and correctness of the produced abstracts, when comparing the abstracts with the original texts. It has traditionally been involved in measuring the similarity between automatically generated abstracts and human prepared ones (e.g., Edmundson, 1969; Kupiec, Pedersen, & Chen, 1995; Hand T.F., 1997; Salton, Singhal, Mitra, & Buckley, 1997). Such an evaluation finds its origin in *information extraction*, which is a text-processing task with similar general goals and processes (see below). The key content of texts is extracted and is compared with a standard human

extraction in terms of quantitatively measuring the completeness (*recall*), correctness (*precision*), superfluity (*overgenerution*), and incorrectness (*fallout*) of the responses (DeJong, 1982; Chinchor, 1992, Chinchor, Hirschman, & Lewis, 1993).

$$\text{recall} = \frac{\text{number of correct responses by system}}{\text{total number of correct responses by expert}}$$

$$\text{precision} = \frac{\text{number of correct responses by system}}{\text{total number of responses by system}}$$

$$\text{overgeneration} = \frac{\text{number of spurious responses by system}}{\text{total number of responses by system}}$$

$$\text{fallout} = \frac{\text{number of incorrect and spurious responses}}{\text{total number of incorrect responses by expert}}$$

In information extraction there is usually agreement on the information to be extracted, nevertheless the evaluation measures employed take into account partial correct responses. The number of correct responses by the system then is augmented with the number of partial correct responses. The latter number receives a weight factor lower than 1 (e.g., 0.5). In text summarization a correct summary is very difficult to establish. If an abstract is manually constructed, different abstracters will produce different abstracts that sometimes have a low overlap in content (Lancaster, 1991, p. 105 ff.; Salton et al., 1997). Especially, when judging critical or interpretive abstracts, agreement on the content is almost impossible. In many cases, it is only possible to judge if the abstracts automatically derived are reasonable or alternatively manifestly inadequate in satisfying the above criteria.

It is also important to evaluate whether the summary meets the user's need and to assess the legibility of an abstract independently from the source text (cf. Sparck Jones & Galliers, 1996, p. 19 ff.). For instance, the content of a summary adequately reflects the desired information, but the readability of the summary severely hampers the task for which the summary is intended. So, an *extrinsic evaluation* is tied to the purpose of the summary and judges the quality of the summary based on how it effects the completion of some other task (e.g., how the abstracts affect retrieval effectiveness when document selection is based upon them). Recently, there have been attempts (Miike, Itoh, Ono, & Sumita, 1994; Mani & Bloedorn, 1997; Tombros & Sanderson, 1998) to develop schemes that measure qualitative features of the systems in a task-based environment. For instance, it is possible to test an increase in speed and accuracy of determining the relevant documents in a list of documents returned by a text search, when the selection is based on text abstracts.

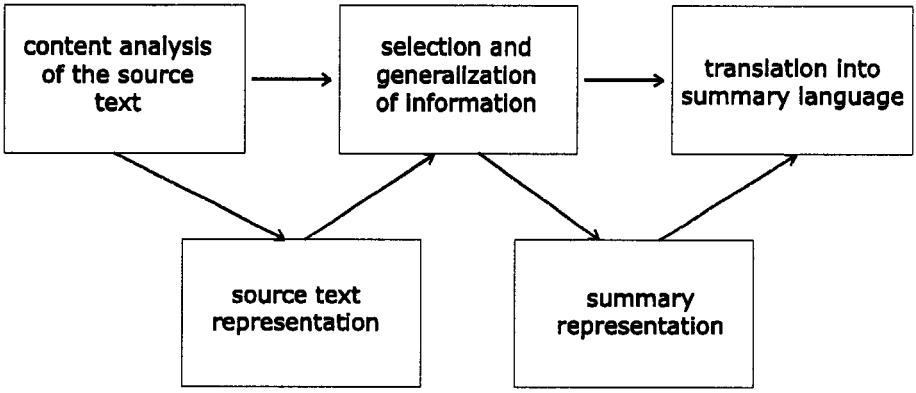


Figure 1. The process of automatic abstracting.

Because many of the summarization tools employ a knowledge-based approach, transportability and maintainability are important to measure (cf. Sparck Jones & Galliers, 1996, p. 159). Transportability measures the cost of porting the system to new applications including the adaptability, conformance, and replaceability of the system. Maintainability of the system concerns its changeability, its testability, and its quality of being analyzable. Because text processing is more complex than when indexing, evaluation of the parsing is also important (Sparck Jones & Galliers, 1996).

Evaluation of summarization tools is often expensive. It is important to carry out enough tests in a sufficiently controlled way. For instance, comparisons with abstracts that are generated by different persons may be necessary (cf. Salton et al., 1997), besides measuring the impact of the abstracts in achieving a text selection task. It is generally agreed that more research is needed to establish adequate evaluation procedures (Sparck Jones & Endres-Niggemeyer, 1995).

3. THE TEXT ANALYSIS STEP

A first group of techniques relies heavily upon knowledge sources to interpret the surface features of a text. These methods find their origin in *natural language processing*, are adequate to generate good abstracts, but are restricted with regard to the application domain. We discuss them under the heading deeper processing. A second group of techniques takes advantages of the word distributions in texts and consists of more shallow

statistical techniques. They originated in *information retrieval* research (indexing with natural language index terms), are weaker in terms of general results, but are more general with regard to the application domain. In between are the methods that use statistical techniques to learn the discourse patterns.

3.1 Deeper Processing

The ultimate goal of the text analysis is the complete understanding of the source text, whereby each sentence is processed into its propositions representing the meaning of the sentence, and whereby the sentence representations are integrated in the global meaning representation of the text. Then, in the transformation step, this representation could be pruned and generalized according to the focus of summary. Current text summarization systems are not so sophisticated, especially in the text analysis step, but nevertheless succeed in creating plausible abstracts. They often combine text analysis with information selection and focus upon finding certain information in the text that is relevant for the abstract. A number of techniques that identify information in natural language texts have been successfully implemented. These techniques rely upon symbolic knowledge and parse the texts guided by this knowledge.

3.1.1 The knowledge

The symbolic knowledge mainly concerns linguistic, domain-specific, and contextual knowledge. The *linguistic knowledge* commonly deals with lexical, syntactical, and semantic properties. It also includes knowledge of discourse structures, especially as flagged by lexical and other surface cues. *Domain world knowledge* deals with the representation of the domain dependent content of the text and is often of semantic nature. The knowledge representation generally integrates the linguistic and the domain modeling. When the text analysis also integrates information selection, *contextual knowledge* that models the *communicative preferences of the users* of the abstract is also needed.

The knowledge is usually captured in *production rules* and *frames*, which are organized into conceptual graphs and semantic networks of frames (see chapter 5). The structured knowledge representations are also called *content schemes*, *scripts*, *templates*, or *text grammars*. A knowledge representation in the form of a content scheme does not only guide the parsing of the text, but also prevails as a target representation of the abstract, which is often generated from the instantiated frames in the schemes or scripts.

Before discussing the common parsing techniques employed in abstracting and information extraction, it is useful to describe common types of grammars (Allen, 1995, p. 19 ff,). The grammar is a formal specification of the structures allowable in the language. A *context-free grammar* is most commonly used to represent the structure of a sentence or of a whole text. A context-free grammar is a treelike representation that outlines how the sentence or text is broken into its major subparts, and how these subparts are broken up in their turn. In the grammar formalism symbols that describe the components of the structure and that cannot be further decomposed are called *terminal symbols*, the symbols that further can be decomposed are called *non-terminal* symbols. For instance, the terminal symbols in a grammar that describes the structure of sentences are the grammatical word classes. The non-terminal symbols can be used recursively making a representation of real nested structures possible. A context-free grammar can describe many structures in natural language and efficient parsers can be built to analyze the texts. In the following example, which describes the structure of the sentence <S> "The judge buried the case.", <NP> (noun phrase) and <VP> (verb phrase) are non-terminal symbols. <ART> (article), <NOUN>, and <VERB> are terminal symbols.

<S> ::= <NP> <VP>
<NP> ::= <ART> <NOUN>
<VP> ::= <VERB> <NP>

A more simple form of grammar is the *regular grammar*. The regular grammars are a subset of the context-free grammars and do not allow non-terminal symbols in their description. Regular grammars are useful to describe lexical patterns in the texts. In the following example a simple arithmetic expression <ARITH> is described with a regular syntax (the letters and arithmetic operators are terminal symbols). The asterisk indicates zero, one, or more repetitions.

<ARITH> ::= ("a" | "b" | … | "z") (("+" | "-" | "*" |"/")("a" | ''b" | … | "z"))*

The most complex structures are recognized by a *context-sensitive grammar*. A context-sensitive grammar takes into account the context of a symbol, i.e., it may have more than one symbol on the left-hand side of the rule, as long as the number of symbols on that side is less or equal to the number on the right-hand side (e.g., <X><Y> ::= <Y><X>).

3.1.2 The parsing techniques

Different *parsing techniques* can be employed when generating a source text representation. The techniques can be distinguished according to the completeness in covering the text, to the kind of grammar used, and to the prevalence of semantic or syntactic components.

The parsing technique is the method of analyzing a text to determine its structure according to the grammar. The parsing of the text ranges from full parsing to text skimming but is in the most cases of text abstracting restricted to a partial parsing (Rau & Jacobs, 1989; McDonald, 1992). A *full parsing* processes every word in the text and allows the word to contribute in the meaning representation of the text. Full parsing needs complete lexical, syntactic, and semantic knowledge of the language of the text to be analyzed and of its discourse properties. Full parsing is applied when the knowledge base fully covers the subject domain (e.g., parsing of a sublanguage). More often, the input text is only partially parsed (*partial parsing*) tolerating unknown elements as much as possible, but relying heavily upon domain knowledge to make up gaps in the linguistic knowledge. Still, the aim is to generate a representation of the text that more or less completely covers its main content. In *text skimming* the parsing is very restricted in order to extract a few pieces of information that are of interest.

In some cases, the information to be included in the abstract and its linguistic context can be described by a *regular grammar*. In these cases, the parsing is often restricted to a *pattern-matching* procedure in which there is a fairly direct mapping from the text to the information to be extracted, without the construction of elaborate intermediate structures. A *finite-state automaton* (see chapter 4) is often employed to recognize the word patterns,

In most cases of text abstracting, a syntactic representation of the sentences or of the discourse structure of the text is used. The parsing then relies upon a *context-free grammar*. There are two major parsing techniques for accessing the text, top-down and bottom-up parsing (Rau, Jacobs, & Zernik, 1989). *Bottom-up parsing methods* usually connect the individual text words into phrases and sentences, and possibly into complete text representations, while instantiating linguistic relations and inferences. Bottom-up parsing is often used for a full parsing of the text. It can perform an in-depth analysis of the text, but needs complete knowledge of the language and subject domain. Even when successful in a text summarization task, this strategy often results in unneeded complexity and inefficiency (Cowie & Lehnert, 1996). *Top-down parsing* is expectation-driven. It is guided by expected structures to be found in the text. It is often used for a partial parsing or skimming of the text. The knowledge in the frames,

templates, or scripts provides the basic semantic units, which have to be appropriately matched in the text by the parsing and these units identify the desired pieces of information in the texts. Consequently, top-down methods are more tolerant of unknown words or grammatical lapses and ignore much of the complexities of the language, but cannot produce any results when the text presents unusual or unexpected structures. The latter is sometimes seen as an advantage for summarization. Since the parsing is unable to understand much of the details, the summary tends to rely on the main structures. So, top-down processing is given precedence in analyzing the source text for text summarization. Both parsing techniques (*top-down* and *bottom-up*) can be *combined* in order to make use of all the sources of information available for the understanding of the subject domain and to make the processing more flexible and robust (Rau et al., 1989; Jacobs & Rau, 1990). Top-down processing can be used to select certain passages in the text that in their turn can be processed in a bottom-up manner in order to produce more detailed content representations.

The majority of the parsers employed in information extraction and text summarization are *semantic parsers* (Hahn, 1990). Although the knowledge structures that guide the parsing process combine lexical, syntactic, semantic, discourse structural, and domain knowledge into a single scheme or grammar, they are essentially semantic, and often only characterize a particular subject domain. They often represent the semantics of individual words and text structures. Parsing natural language is achieved by mapping the utterances directly into semantic representation structures without considering a logically separated, intermediate level of syntactic representations. Although syntactic language structures can provide auxiliary information for semantic interpretation (e.g., syntax of individual sentences and phrases, syntax of ordering of discourse segments), the primacy of semantics over syntax is determined by the ultimate goal of determining a meaning of the natural language utterances. The result of the parsing is a semantic, rather than a syntactic description of the text.

The “deeper” techniques of text analysis rely on a set of expectations about the contents and surface features of the texts. They are successful for analyzing texts of which the discourse characteristics are predictable and well understood (e.g., news stories, financial, and commercial reports) or for identification of specific information that can be found in predictable contexts. In the next sections, the techniques are illustrated. The applications show a definite trend from systems that heavily rely upon knowledge of the subject domain towards systems that besides domain knowledge incorporate knowledge about discourse structures.

3.1.3 The original models

The earliest models in text abstracting and information extraction from texts come from Sager (1975), Rumelhart (1975, 1977), Schank (1975; Schank & Abelson, 1977). Sager employs a *sublanguage grammar* to extract information from medical texts, The grammar is strongly based upon the semantics of the sublanguage domain. Rumelhart proposes the idea of so-called *story grammars* for understanding and summarizing texts, which describe the discourse properties of particular text types. He analyzes stories into hierarchical structures that reflect the schematic and rhetorical structure of the text type. Schank defines all natural language words in terms of elementary primitives or predicates in an attempt of capturing the semantic content of a sentence. A *conceptual dependency representation* specifies the action of the sentence (e.g., as reflected by the verbs of the text) and its arguments (*semantic case roles*) (e.g., agent, object). The representations are ordered in a *script*, which outlines sequences of events or actions.

An early successful system is *FRUMP* (DeJong, 1977, 1982). FRUMP summarizes typical newspaper stories (kidnaps, acts of terrorism, diplomatic negotiations, etc.). It accurately extracts specific information (e.g., nature and location of an event) from stories in the pre-selected topic areas and generates a summary based upon the extracted information. The knowledge is organized in "sketchy scripts", which contain a priori expectations about the subject domain. A script is represented in a top-down structural form. It describes sequences of events, thus imposing a kind of structure upon the stories, and inferences about additional events that may occur. Some syntactic knowledge is present to determine the general sentence location of an expected word. A parser skims through texts looking for words signaling a known script, for which it is able to predict or expect the occurrence of other words or phrases, and so builds up the outline of a story. It is only interested in and only interprets these parts of the text that relate directly to elements of the script, the rest of the text is ignored or skipped. FRUMP is a model of specialized abstracting where only those parts of texts are analyzed and recorded which are of interest for a specific task tailored to specific users' needs. When evaluated, FRUMP understands more than 50% correctly, i.e., understanding everything from the story that its script predicts. FRUMP has been used to create abstracts employed for query matching and an increase in both recall and precision in retrieval effectiveness has been noted (Mauldin, 1991).

The system of Lehnert (1982) identifies plot units in stories of which the plot structure is not fixed. Plot units have the form of propositions and are composed of affect states (e.g., positive events, negative events, mental

states) that are linked by four types of relation (motivation, actualization, termination, and equivalence). The recognition of affect states is based on a large predictive knowledge base containing knowledge about plans, goals, and themes. The analysis of a story in terms of plot units results in a complex network where some units are subordinated to others. Summarization of a story involves essentially the extraction of top-level plot units.

3.1.4 Other applications

Many other information extraction systems that generate an abstract of the text have been described (Tait, 1985; Fum, Guida, & Tasso, 1985; Berrut & Chiaramella, 1989; Ciravegna, 1995; and see Hahn (1989) and Jacobs (1992) for overviews of other systems). Famous systems are TESS (Young & Hayes, 1985), SCISOR (Jacobs & Rau, 1990), CONSTRUE (Weinstein & Hayes, 1991; Hayes, 1992), and FASTUS (Appelt, Hobbs, Bear, Israel, & Tyson, 1993). Many systems are evaluated and compared to human benchmarks in the semiannual *Message Understanding Conferences* (*MUCs*) and in the *ARPA's Tipster Text Program*, which coordinates multiple research groups and agencies, both sponsored by the US government. In terms of recall and precision the performance of the MUC and Tipster systems is characterized by an average of about 40% recall and 50% precision measured in terms of correspondence between information intellectually extracted and the one automatically generated (MUC-4, 1992; Cowie & Lehnert, 1996). But, there is a large variety in performance results. The performance of the individual systems is largely similar, but some information is much more difficult to extract from texts than others is. In terms of speed, machine performance far exceeds human performance.

3.1.5 The significance of discourse structures

Work on information extraction is important for the analysis and selection step in text summarization. However, the systems described above contain a bulk of domain-dependent knowledge. As it is explained in chapter 2, communication by means of natural language text (spoken or written) is governed by *discourse patterns.* It is acknowledged that knowledge of these patterns is indispensable in text understanding, even if this understanding is only partial, as it is often the case in abstracting the content of text (Moens, Uyttendaele, & Dumortier, 1999b). In general, knowledge of discourse structure is much less domain dependent. Some structures, such as the text-type dependent superstructure are sometimes exclusively used in a certain

text typology, but many other communication structures are widely used. Hence, there is an emerging interest in using discourse patterns in text abstracting. This interest is not new. Early text summarization systems already employed discourse patterns in a limited way.

The *schematic structure or superstructure* of a text and its signaling linguistic cues have always been recognized as being significant in text summarization (Luhn, 1958; Maeda, Momouchi, & Sawamura, 1980; Paice, 1991). For instance, titles and subtitles are supposed to conceive the content of a text (Bernstein & Williamson, 1984; Paice & Jones, 1993). More elaborated schematic structures are used to summarize news stories (Liddy, McVearry, Paik, Yu, & McKenna, 1993). The text type dependent superstructure and the text type independent *rhetorical structure* are often hinted by typical natural language expressions in the text. In early summarization systems *cue words and indicator phrases* are used to indicate significant *sentences* in a text or to reject sentences that are without any value in the abstracting process (Edmundson, 1969; Rush, Salvador, & Zamora, 1971; Paice, 1981). Rhetorical cues continue to be highly valued in present summarization systems (Miike et al., 1994; Brandow, Mitze, & Rau, 1995). Rhetorical relations, especially as flagged by lexical and other surface cues, are seen as standard, known ways of organizing text that are conventionally associated with achieving certain communicative effects. Also, other rhetorical markers provide cues to relevant text fragments (italics, bolds, underlining, and orthographic markers). Presently, taxonomies of discourse segment relations and their signaling linguistic phenomena are at the disposal of text summarization (Paice, 1990; Mann, Matthiessen, & Thompson, 1992; Hovy, 1993b).

Not surprisingly, the *thematic structure of a text* is important in automatic abstracting its content. In early systems, it was recognized that the *first* or *the last sentence of a paragraph* and *sentences at the beginning or end of a document text* are usually the most central to the theme of the text (Baxendale, 1958; Edmundson, 1969). Kieras (1985) and Kupiec et al. (1995) confirm these findings. Such locational cues are useful to identify sentences to be included in the abstract. Specific words and phrases cue thematic content or shifts in thematic content. The TOPIC system of Hahn (1990) is a good example of exploiting the thematic structure of a text for summarization. Hahn implemented three basic patterns of *thematic progression* in texts: the elaboration of one specific topic within a text passage, the detection of topic shifts within a sentence, and deriving the topic, which is composed of different subtopics, across text passages. But, topic recognition in TOPIC still strongly relies on specific domain knowledge, embodied in lexical experts and frames. The *thematic structure*

of individual sentences can be exploited to pinpoint topics that are most into focus, which can be used for identifying key text topics and sub-topics (Sidner, 1983; Kieras, 1985). Here also, cue words hint significant concepts (Paice & Jones, 1993) or provide the context for the thematic roles of certain phrases and clauses (Wendlandt & Driscoll, 1991). According to Kieras (1985), topic processing can proceed largely on the basis of limited knowledge of the semantics of the subject matter without an understanding of the passage content. This is a hypothesis that should be tested in practical systems. The thematic structure of texts is especially useful for generating abstracts that reflect the main topics and subtopics of texts. Knowledge of this structure is also important for producing abstracts at different levels of topic granularity.

The usefulness of discourse patterns in text summarization awakes the idea of representing texts by means of a *text grammar* (Paice, 1981, 1991; Paice & Jones, 1993; Rama & Srinivasan, 1993). Texts like sentences have a kind of grammar, i.e., a set of implicit rules that writers and readers assume, that help govern the selection and ordering of elements in a discourse, and that make texts understandable to one another (cf. Reichman, 1985). The importance of discourse structure in text summarization was also stressed at the *Spring Symposium on Intelligent Text Summarization* (1998) organized by the *American Association for Artificial Intelligence.*

3.2 Statistical Processing

There is also a remarkable interest in using statistical techniques in text abstracting. The statistical techniques are shallow, in the sense that they severely reduce the domain and linguistic knowledge needed for the analysis of the source text or learn this knowledge. Consequently, these techniques are more independent from the subject domain and text genre, and can be broadly applied. A major approach concerns the identification of important topic terms and the extraction of contextual sentences that contain them. An automatic structuring of the text according to its topics is important. Another statistical approach regards techniques that classify the discourse parameters involved in text summarization based on example abstracts of example source texts.

The idea of using statistical techniques in text summarization also goes back to Luhn (1958). At that time, automatic abstracting and text indexing were strongly related (Baxendale, 1958; Earl, 1970). Statistical text summarization has received renewed interest and was a theme of the *Workshop on Intelligent Scalable Text Summarization* (1997) organized by the *Association for Computational Linguistics.*

3.2.1 Identification of the topics of a text

In information retrieval research there is a long tradition of identifying *words and phrases* in a text that *reflect* its topics based on their distribution characteristics in the text and/or in a reference corpus (see chapter 4). The topical terms can form the basis of the text's summary.

Significant words and phrases reflect a text's content and may serve well as crude abstracts (*keyword abstracts*) (Cohen, 1995). Phrases, especially noun phrases, are considered as important semantic carriers of the information content (Maeda et al., 1980; Kupiec et al., 1995). There are many techniques for word and phrase weighting in texts. Moreover, significant words and phrases help in identifying the *relevant sentences* that are retained for summary purposes. A simple, but still attractive approach extracts sentences that contain highly weighted terms possibly in close proximity (Luhn, 1958; Edmundson, 1969; Earl, 1970; Salton, 1989, p. 439 ff.). So, clusters of significant words within sentences are located and sentences are scored accordingly. A variant hereof considers query terms as the content terms around which the summary is built, i.e., highly weighted query terms in close proximity determine the sentences to be extracted (Tombros & Sanderson, 1998). This variant allows the summary being tailored to the need of a user.

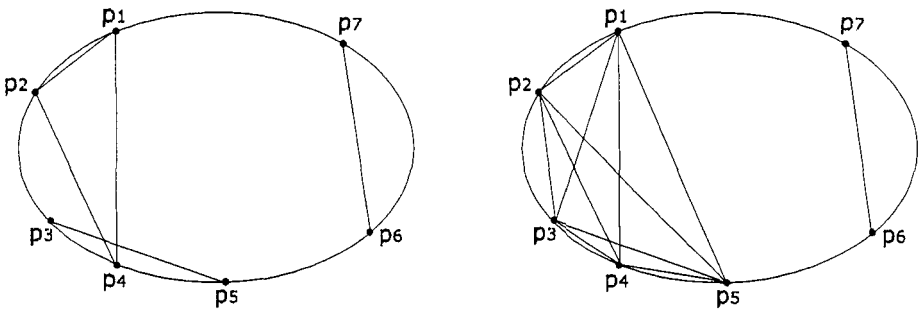


Figure 2. Paragraph grouping for theme recognition: A lower similarity threshold connects more paragraphs into a broader theme group.

Significant words and phrases also help in determining the thematic structure of a text and in extracting representative sentences or paragraphs of important text topics to form a summary.

There is a growing interest in identifying the *thematic structure of a text* based on its term distributions (Figure 2). Techniques concern the grouping of textual units (fixed number of words or units marked orthographically such as sentences or paragraphs) that have similar patterns of content terms. This approach has been elaborated by *Salton* and his co-researchers (Salton, Allan, Buckley, & Singhal, 1994; Salton et al., 1997). In their approach, paragraphs are grouped if there is sufficient overlap of their content terms. Such a grouping may reveal the main topics of the text. The similarity between a pair of paragraphs is computed by applying the cosine function upon the vector representations of their content terms (cf. Jones & Furnas, 1987). Paragraphs are grouped if their mutual similarity exceeds a predefined threshold value. A threshold similarity value allows broadening or narrowing the grouping ideally allowing for hierarchically arranged contexts wherein users can zoom from one context to another (Salton et al., 1997). A similar course follows the research of Hearst and Plaunt (1993) (Hearst, 1997), which aims at detecting the subtopics of a text. Based upon the assumption that the main topics of an expository text occur throughout the text, and the subtopics only have a limited extent in the text, their system *TextTiling* automatically reveals the *structure of subtopics*. TextTiling computes the similarity of each two adjacent text units. The text units compared consist of about 3-5 sentences. The resulting sequence of similarity values is placed in a graph. The graph is smoothed and examined for peaks and valleys. Valleys in the graph identify ruptures in the topic structure. TextTiling has been applied to structure articles of a scientific journal according to subtopics.

Once the thematic structure is determined, it can be used to selectively *extract important sentences or paragraphs* from the text and traversing the extracted units in reading order to construct a text extract that serves as a summary. The idea goes back to Prikhod'ko and Skorokhod'ko (1982), who studied the importance of links between sentences in text summarization. Each sentence is scored by the *number of links* (common content terms or concepts) with the other sentences of the text. Sentences the score of which surpasses a threshold are included in the abstract. This approach is based on the assumption that sentences related to a large number of other sentences are highly informative and are prime candidates for extraction. Recently, a few algorithms have been proposed to extract representative text paragraphs in order to form a readable and topically balanced abstract (Salton et al., 1997). The algorithms suggested have relatively poor results. When

compared with manual abstracts an overlap of maximum 46% is obtained. A best score is achieved with an algorithm that extracts paragraphs that are highly linked with other paragraphs or have a large overlap in terms of content terms with other paragraphs. In chapter 8 we discuss the shortcomings of these algorithms and propose alternative ones.

3.2.2 Learning the importance of summarization parameters

Discourse patterns, including the distribution and linguistic signaling of highly topical sentences, may vary according to the document corpus or the text type. Also, when information is selected from the source to be included in a specific task-oriented summary, discourse patterns can have different weights. There are experiments in *learning* the *value of discourse parameters*. Kupiec et al. (1995) compute the weight of certain discourse patterns based upon an example text base and their abstracts. On the basis of a corpus of technical papers with abstracts written by professional abstractors, the system identifies those sentences in the text which also occur in the summary. It then acquires a model of the "abstract-worthiness'' of a sentence as a combination of a limited number of properties or parameters of that sentence. Properties that are accounted for are: length of sentences, sentences containing indicator phrases, or sentences following section headings that contain indicator phrases, sentences in the first ten and the last five paragraphs, the first, final or medium sentences, sentences with frequent content words, and sentences with proper names that occur more than once. A classification function (*Bayesian independence classifier*) (cf. chapter 5 (12-15)) is developed that estimates the probability that a given sentence is included in the abstract given the probability of its properties in the texts of the training base. Each sentence is described by a number of discourse patterns and the probability of inclusion in the summary is computed based on estimates of the probabilities of the patterns in example abstracts and example source texts. When abstracting a new text, its sentences are ranked according to this probability and a specified number of scoring sentences is selected. This approach offers a direct method for finding an optimal combination of selection heuristics based on discourse patterns. The summarizer has been tested on publications in the scientific-technical domain. The best results (43% correctness in correspondence with manually extracted sentences by professional abstractors) are obtained by a combination of location, cue phrase, and sentence length heuristics. The experiment is replicated by Teufel and Moens (1997), who demonstrate the usefulness of the approach for text analysis and selection in a summarization task. It might be noted that statistical independence of discourse patterns

employed in a Bayesian classifier is sometimes a false assumption. Recent attempts to use *discriminant functions* and techniques for *inducing logical rules* (e.g., C4.5 algorithm of Quinlan, 1993) (see chapter 5) in acquiring discourse patterns show encouraging learning performance (Mani & Bloedorn, 1998).

From the above, it is clear that the statistical techniques offer opportunities to develop unsupervised as well as supervised techniques to learn discourse patterns and to avoid, at least partially, the knowledge acquisition step in text analysis. This is a promising research area. Parallel to this research, more must become known about the discourse patterns of source texts and about significant discourse parameters for text summarization.

4. THE TRANSFORMATION STEP

4.1 Selection and Generalization of the Content

The first step of text summarization regards text analysis of the source text and results in a representation of its content. In a second step, this representation will be pruned and condensed in order to form a summary representation. Summarization always involves selection and generalization of the content of the text. This transformation step requires additional knowledge about the task and audience of the abstracts to guide the selection of the information and about the subject domain to conduct an accurate generalization of the information. Selection and generalization are very important when summarizing multiple texts in one summary.

Selection of relevant information is highly tied to the discourse structure of the original text. For instance, when the abstract must reflect the main topics of the text, the thematic structure is important in order to select the right information. Or, certain segments of the superstructure convey valuable information to be included in the summary. As seen above, many practical systems combine text analysis and information selection and restrict the summarization process to the extraction of sentences that have a high relevancy score.

A still more difficult task to perform automatically is the *generalization* of the selected information. Generalization is condensing the information to a more abstract form. For instance, it regards deriving from a description of girls playing with dolls and boys playing with trains the description of children playing with toys. This task requires a bulk of semantic information.

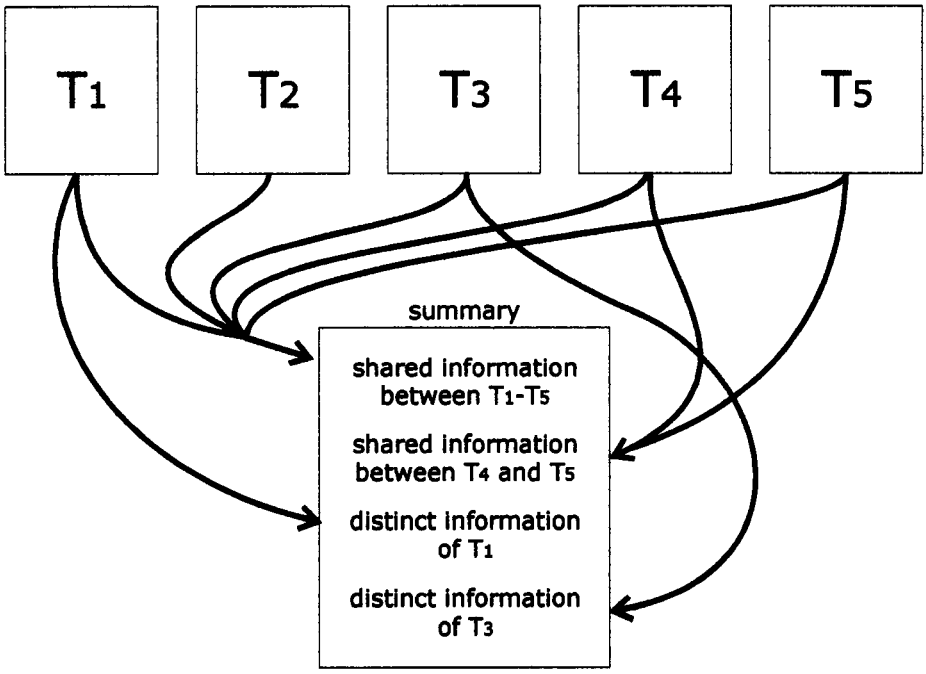


Figure 3. Summarization of multiple texts (T_1 ... T_5).

Thesauri and ontologies with semantic classifications of the lexical items are certainly needed (cf. McKeown & Radev, 1995), but probably not sufficient to describe the selected information at a more general level. Hahn (1990) derives the main topic of a text from its subtopics by instantiation of generalization/classification relations holding among the subparts of a frame hierarchy in the domain knowledge base.

Selection and generalization of information also control the *length of the summary*, i.e., the degree of compression of the original text. The length of a summary in proportion to the length of the original text can vary (see chapter 3, p. 60). Ideally, one should be able to zoom in and out on informational detail of the abstract. But, this is probably a long-term goal.

4.2 Selection and Generalization of the Content of Multiple Texts

Selection and generalization of information are important steps when generating a summary representation from *multiple source representations*.

When merging different source representations (e.g., the merger of similar fields of instantiated templates or frames) the focus of the final summary determines the selection of information in the individual source representations. A particular challenge is to summarize the similarities and differences in information content between these documents (Figure 3) in a way that is sensitive to the need of the users. For instance, when information in texts changes over time, it might be important to merge and generalize the stable information in the texts and to identify the most recent dynamic information to form the summary representation. The research of McKeown and Radev (1995) focuses on the generation of abstracts of multiple news articles on the same event. Their system attempts to generate fluent text from sets of templates that contain the salient facts reported in the input texts, which are extracted from them by the ARPA message understanding systems. The research focuses on techniques to summarize how the perception of an event changes over time, using multiple points of view over the same event or series of events. Text fragments are compared for change of perspective, contradictory statements, gaps in the information, additions, confirmations, and refinements of the information. This research indicates that comparative abstracts are not completely out of reach, but requires source text representations of good quality, which is currently not always accomplished except for restricted subject domains. This finding is confirmed by Mani and Bloedorn (1997). Salton et al. (1997) discuss the possibility of extracting text units from multiple texts, which are semantically linked based on vocabulary overlap, in a way similar to extracting text units from groups of related text paragraphs. Experiments must confirm the usefulness of such an approach. It is known that in a document corpus vocabulary is much more diverse than within a single text, and the problem of synonymy and ambiguity is prominent.

5. GENERATION OF THE ABSTRACT

Once a summary representation is built from the source representation, the final abstract is generated based upon the summary representation. The complexity of this task depends on the format of the desired abstract. For instance, when the abstract is a text profile stating extracted information in the form of well-defined semantic fields, the task of summary generation is nearly absent. On the other hand, it is a more complicated task to generate a summary that forms a complete, coherent, and comprehensible text, which is comparable to most manually created abstracts. This requires additional, mainly linguistic knowledge and techniques developed in the field of text

generation. In between a profile and a perfectly coherent text, there are the many practical systems that perform some kind of editing of the sentences or, of other text units extracted from the source text.

Ideally, the summary is built from the propositions or from the semantic fields in the summary representation. Then, the next step involves *text generation*. Text generation is a broad field and can range from the selection of information to be communicated and text organization, to the generation of linguistically well formed surface expressions such as sentences and the lexical choice of the words employed (McDonald, 1993). The purpose of text generation tools lies in a better readability of the text in order to enhance its communicative value. Discussion of text generation tools is beyond the subject of this book. Horacek & Zock (1993) give a good overview of this subject. Summary generation imposes an additional constraint: The content has to fit into a minimum of text lines (McKeown, Robin, & Kukich, 1995).

When *sentences* are *extracted* from a text and reproduced in reading order in the summary, the result is not always a fluently readable text because of a lack of coherence and of disturbance by other factors. This is less a problem for extracted paragraphs, which are often coherent units on their own. Extracted sentences often need a limited form of text editing.

1. The most important problem regarding summary coherence is the frequent presence of "*dangling*" *anaphoric* and *cataphoric references* (Tait, 1985; Paice, 1990; Paice & Jones, 1993). For instance, pronouns, demonstratives and comparatives used in the sentences may only be understood by referring to an antecedent appearing earlier (anaphoric reference) or occasionally later (cataphoric reference) in the text. Ellipses cause similar problems. Anaphoric references receive a great deal of attention. Simple solutions regard the deletion of sentences with anaphoric references (Paice, 1990) or the addition of a preceding sentence to the one that contains the anaphor (Rush et al., 1971). The former endangers the coverage of the information. The latter does not guarantee that the anaphor is resolved: It may refer to earlier sentences in the discourse. Sometimes, inclusion of previous text units possibly up to a specific cue term is suggested (Paice, 1981). Determining the correct antecedent requires a linguistic analysis in which discourse structure plays an important role (Grosz, 1981; Sidner, 1983) and is a problem to be solved at the text analysis step (Bonzi & Liddy, 1989).
2. Other *rhetorical connectives* possibly distort the readability of a summary that is composed of extracted sentences (Paice, 1990). Among them are cohesive features that indicate the nature of relationship between a sentence and its predecessor or successor. For instance, when the extracted sentence forms the contrast to a previous text passage

indicated by the rhetorical cue "on the other hand" at the beginning of the sentence, it will contrast the previous sentence of the summary, which not always complies with the content of the source text. Deletion of such rhetorical cues in the extracted sentences or inclusion of a previous sentence again is a weak tentative to solve the problem. Real solutions can only come from considering discourse structure in text analysis.

3. Also references to tables, figures, or other texts, and material in parenthesis are not always appreciated in the summary. They can be easily deleted (Mathis, Rush, & Young, 1973; Paice, 1990).
4. It is important that the summary is a *concise description* of the content of the original document without a loss of clarity. Linguistic knowledge can help in merging sentences with repetitious structure and coordinate sentences with conjunctions (Mathis et al., 1973). On the other hand, it is often necessary to augment very short sentences with a neighboring sentence to increase the clarity of the summary.

Despite its benefits in terms of readability of the abstract, *linguistic reformulation is not always desired*. First, when the abstract is intended for rapid reading in a text selection task, its editing is sometimes not necessary and only slows down its creation (Salton et al., 1997). Second, it is sometimes important that the summary follows as much as possible the wording of the original text (Endres-Niggemeyer, 1989). For instance, it is appropriate to include original text sentences in the summary of legal texts, because the danger of misinterpretation is large when altering the original formulations (Uyttendaele, Moens, & Dumortier, 1998).

6. TEXT ABSTRACTING: ACCOMPLISHMENTS AND PROBLEMS

Often, an unordered set of index terms cannot accurately represent the content of texts. The richer semantic representation of an abstract compared to the "bag of words" representation in case of indexing has definite advantages despite its more complex computation (see chapter 3). The value of automatic text abstracting is not questioned, but it is important to create abstracts that are true reflections of the content of texts and that are useful in the task that they are intended for. This is still problematic (cf. Edmundson, 1964), but there are promising directions to pursue.

1. The field of *information extraction* offers valuable solutions to identify information in texts. However, the proposed solutions heavily depend on

external knowledge, especially domain knowledge. Because of the knowledge acquisition bottleneck, successful applications operate in restricted subject domains, When knowledge of the domain model is cheap to acquire manually, this approach can be encouraged (cf. Cowie & Lehnert, 1996). But, there is an emerging interest to concentrate upon generic knowledge in text abstracting such as linguistic knowledge and especially the discourse patterns of whole texts. Classical natural language parsing of the text might not yield a complete understanding of the text, but it may yield enough predictions for text abstracting. More importantly, discourse analysis has explored many discourse phenomena for spoken dialogue as well as written text, and has in particular proposed models of text structure and structural relations that appear especially relevant to summarizing. Research on text typology clearly is significant, either because different genres may require different abstracting strategies, or perhaps there are general genre-independent abstracting strategies to be discovered. Discourse structure is important for locating salient pieces of information in texts. It is generally agreed that we need more linguistic and social-cultural studies on the nature of discourse and text (cf. Endres-Niggemeyer, 1989; Sparck Jones & Endres-Niggemeyer, 1995; Moens et al., 1999b). Besides integrating more generic knowledge, it is also important to develop tools that facilitate the implementation of knowledge across subject domains and text typologies. Finally, there is the interest in automatically acquiring the linguistic, especially the discourse patterns, as well as the domain-dependent knowledge. The learning techniques form a bridge with the more general statistical techniques of the next paragraph.

2. The discipline of *information retrieval* traditionally exploits *statistical techniques* for content identification in texts of broad, unrestricted domains. Additionally, some of the techniques that are recently developed for recognition of thematic structures in texts have a potential for automatic text abstracting. Also, the supervised techniques for learning classification patterns are promising. They are useful for acquiring the typical discourse patterns of document collections, or for learning domain concepts.
3. The transformation step in which a source text representation is reduced to form the summary representation is too often restricted to an information selection process. Replacing the concepts of a source text by more *general concepts* in the summary text is rather neglected in automatic text abstracting.
4. *Human summarizing* as a professional activity has practices and guidelines that are useful as a source of summarizing models. For

instance, psychological studies of discourse reading and its retention in memory as evidenced by summarizing, can throw further light on text features that are remembered or on the properties of a text that serve to identify what is important to it.

5. There is a growing interest in *abstracts of multiple texts*. Comparative abstracts of multiple texts are especially beneficial for accessing large document collections. Here, it is very important to start from good source representations of the original texts. Additionally, developing statistical techniques that recognize similarities and differences between the source representations is a promising research area.
6. In order to tailor abstracts to specific needs, we need more studies about how abstracts can be used in text retrieval and other related text-based tasks and how the use determines form and content of the summary (cf. Sparck Jones & Endres-Niggemeyer, 1995).
7. A final problem concerns *evaluation* of the generated abstracts. More research is needed to develop suitable effectiveness measures.

7. CONCLUSIONS

Summarization is crucial to information and knowledge organization. Automatic abstracting is a good solution for *managing a textual information overload*. The abstract that is automatically generated, despite being an approximation of the ideal one, is very valuable in document and information selection from large collections.

In this chapter we emphasized text analysis and selection of salient information from the original text. The techniques fall into two classes. There are the ones that rely heavily on symbolic knowledge, produce good quality abstracts, but are often tied to a specific application. On the other hand, there are the more general techniques that statistically process distribution patterns of words, but produce less accurate abstracts. Learning techniques bridge the gap between the domain-specific stronger methods, and the more general, but weaker methods. Several research strategies have been proposed, some of which will be explored in the following chapters. On one hand, we need more studies of discourse and text in order to generate cohesive, properly covered and balanced abstracts at different levels of informational detail. On the other hand, the development of statistical programs for pattern recognition is important for acquiring the discourse patterns, especially the domain- and/or collection dependent text patterns. These techniques might include supervised as well as unsupervised learning algorithms.

PART III

APPLICATIONS

This page intentionally left blank.

Chapter 7

TEXT STRUCTURING AND CATEGORIZATION WHEN SUMMARIZING LEGAL CASES

1. INTRODUCTION

Computers become prominent in law courts and offices of public prosecutors. As a result a huge amount of electronic texts is available. There is, however, an urgent need for intelligent tools that make the information in legal texts manageable (Susskind, 1996, p. 107 ff.). This information is useful for different legal professionals and in different applications. It can be used as indices in search engines that retrieve or route texts, as key extracts that make it easier for selecting documents, or as direct answers in question-answering systems. In more advanced applications this information is effective as straight knowledge input in expert systems or as case features in case-based reasoning systems.

The SALOMON (Summary and Analysis of Legal texts f0r Managing On-line Needs) project developed and tested several techniques to make a vast corpus of Belgian criminal cases (written in Dutch) easily accessible. A system is built that automatically extracts relevant information from the fill-text of a case, and uses it to compose a summary of each decision. The summary has the format of a case profile ("index card"), which facilitates the rapid determination of the relevance of the case (cf. *indicative abstract*). Its user is informed of the name of the court that issued the decision, the decision date, the offences charged, the relevant statutory provisions disclosed by the court, and the important legal principles applied. Moreover, the summary can act as a case surrogate in text search (cf. *informative*

abstract). SALOMON is a test case for the long-term goal of making the totality of criminal jurisprudence comparable on a national level, hereby increasing the value of the information in the cases. In addition, the project contributes to the study of more general methods for text classification, information extraction and text summarization.

In a first step, the case category, its major semantic components, general information (e.g., date, court name, relevant legal foundations) and the non-relevant paragraphs of the text are identified using a text grammar approach. In a second step, relevant paragraphs of the offences and of the opinion of the court are further abstracted. It is the first step of the project that is the subject of this chapter. The second step is the subject of the next chapter.

This chapter is organized as follows. We describe the text corpus and the desired output of the system. The methods are discussed and evaluated. We finish with summing up the contributions of the study. A more detailed description of the research is given in Moens and Uyttendaele (1997) and in Moens, Uyttendaele, and Dumortier (1997).

2. TEXT CORPUS AND OUTPUT OF THE SYSTEM

An expert in criminal law studied a sample of Belgian criminal cases (Uyttendaele, Moens, & Dumortier, 1996, 1998). This analysis resulted in a detailed description of the categories, structure, and the parts of the case that are relevant to include in its summary.

Belgian criminal cases can be classified into *7 main categories*, distinguishing general decisions from particular ones. The latter concern appeal procedures, civil interests, refusals to witness, false translations by interpreters, infringements by foreigners, or the internment of people.

The criminal cases have a *typical form of discourse* (superstructure). They are made up of 9 ordered elements, some of which are optional:

1. *superscription*, containing the name of the court and the date;
2. identification of the *victim*;
3. identification of the *accused*;
4. *alleged offences*, describing the crimes and factual evidence;
5. *transition formulation*, marking the transition to the grounds of the case;
6. *opinion of the court*, containing the arguments of the court to support its decision;
7. *legal foundations*, containing statutory provisions applied by the court;
8. *verdict*;
9. *conclusion*, possibly containing the name of the court and the date.

Some of these components have an interesting substructure (e.g., date and name of the court in the superscription, irrelevant paragraphs in the alleged offences, irrelevant paragraphs in the opinion of the court, irrelevant foundations in the legal foundations). In total we defined 14 different case components or segments relevant for abstracting purposes, some of them being subsegments of larger text segments. The segments present themselves in the text as: text blocks delimited or categorized by typical word patterns (e.g., the transition formulation), texts blocks preceding and/or following another text segment (e.g., identification of the victim), text paragraphs delimited or characterized by typical word patterns (e.g., irrelevant paragraphs in the alleged offences), text sentences delimited or characterized by typical word patterns (e.g., irrelevant foundations), or plain word patterns (e.g., name of the court). A word pattern is a combination of one or more text strings.

The most relevant parts of a case are the alleged offences, the opinion of the court, and the legal foundations. The *alleged offences* give a description of the crimes a person is accused of. The *opinion of the court* allows distinguishing three types of cases within the studied corpus: *routine cases* (containing only routine, unimportant grounds in their opinion), *non-routine cases* (containing other than routine-grounds), and *leading cases* (containing more than 5 "principle grounds"). Principle grounds are the paragraphs of the opinion in which the court gives general, abstract information about the application and the interpretation of some statutes. The routine and the leading cases represent 35% to 40% and 3% to 5% of the total corpus respectively. In the non-routine cases, the judge elaborates the crime themes, taking into account the factual evidence and, in case of leading cases, the application of specific statutes. The *legal foundations* consist of a complete enumeration of legal texts and articles applied by the court. Several of these foundations (*routine foundations*) are cited in each case, while others concern the essence of the case.

After examining *intellectually constructed headnotes* of printed law reports, it was decided that it would be interesting to extract the following information from the case:

1. The *name of the court* that pronounced the decision;
2. the *date of the decision*;
3. the key paragraphs that describe the *crimes* committed;
4. the key paragraphs and terms that appear to express *the essence of the opinion of the court*;
5. references to the applied non-routine *foundations*.

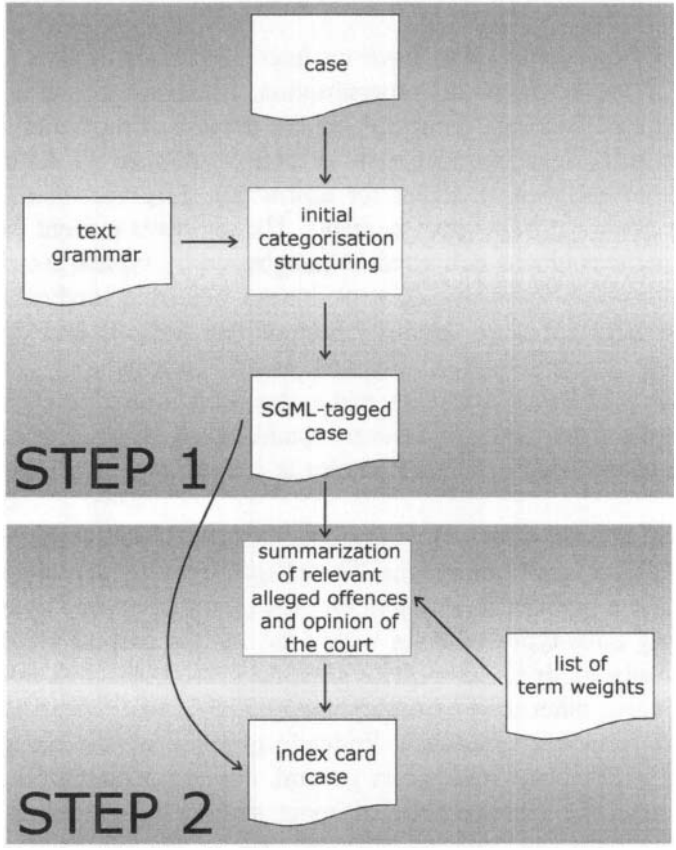


Figure 1. Architecture of the SALOMON demonstrator.

To realize the goals, a demonstrator (Figure 1) is built in the programming language C on a Sun™ SPARC station 5 under Solaris® 2.5.1. It produces a summary ("index card") of a criminal case.

The expert in criminal law interviewed other experts in the field and people responsible for the publication and manual summarization of cases in professional journals. When intellectually abstracting, an initial step regards the identification of the case category, of semantically relevant components, and of insignificant text segments. Similarly, our automatic abstracting procedure consists of *two steps*.

The *first step* identifies the general category and the structure of the case. Also, irrelevant parts of the text of the alleged offences and opinion of the court, and routine foundations are identified. Here, the linguistic context of the information is predictable and the cases are processed based upon a *representation of the texts* that captures the syntax and semantics of the discourse. The result of the initial categorization and text structuring is a

case tagged in SGML(*Standard Generalized Markup Language*)-syntax. A head tag marks the general category of the case. The identified text segments are marked with the appropriate category tags. Some records on the index card (such as date, name of the court, and non-routine legal foundations) can be readily extracted from this structured text. In the *second* abstracting step, the system further summarizes the relevant parts of the alleged offences and the opinion of the court. The remainder of this chapter discusses the first step of the abstracting process.

3. METHODS: THE USE OF A TEXT GRAMMAR

3.1 Knowledge Representation

A text is usually composed of different components or *segments* which fulfill its communicative goals and which are combined according to specific semantic relations. They may concern paragraphs, sentences, or more informal text blocks of varying length. The text segments may be classified and/or delimited by linguistic and domain clues, which are white space characters or punctuation marks, and/or word patterns. Some of these segments are relevant for categorizing, indexing, or abstracting purposes.

The formalism that we designed allows representation of the major semantic units of a text, their attributes, and relations in the form of a text grammar. The formalism represents the text grammar as a *semantic network of frames*. The nodes of the network represent the objects with their attributes, the lines the relations between the objects. Frames offer the possibility to describe complex objects in a detailed way by treating a cluster of information as one entity. Frames can be reused. Frames can be organized in a network, reflecting document structure and content.

A *segment frame* defines a text segment: Its slots describe the segment and its attributes. Each segment has a name (category), which may indicate its communicative goal. Segments belong to one of the following segment types: limits, paragraph, sentence, phrase, and word pattern. A “limits” segment is a text block delimited by word patterns (e.g., indicator words or phrases) or by other segments and possibly characterized by word patterns. The complete text is a special case of this segment type and may be delimited by the beginning and the end of the text file. A “paragraph” segment is delimited by one or more new line characters, and possibly defined by delimiting or classifying word patterns. A “sentence” segment

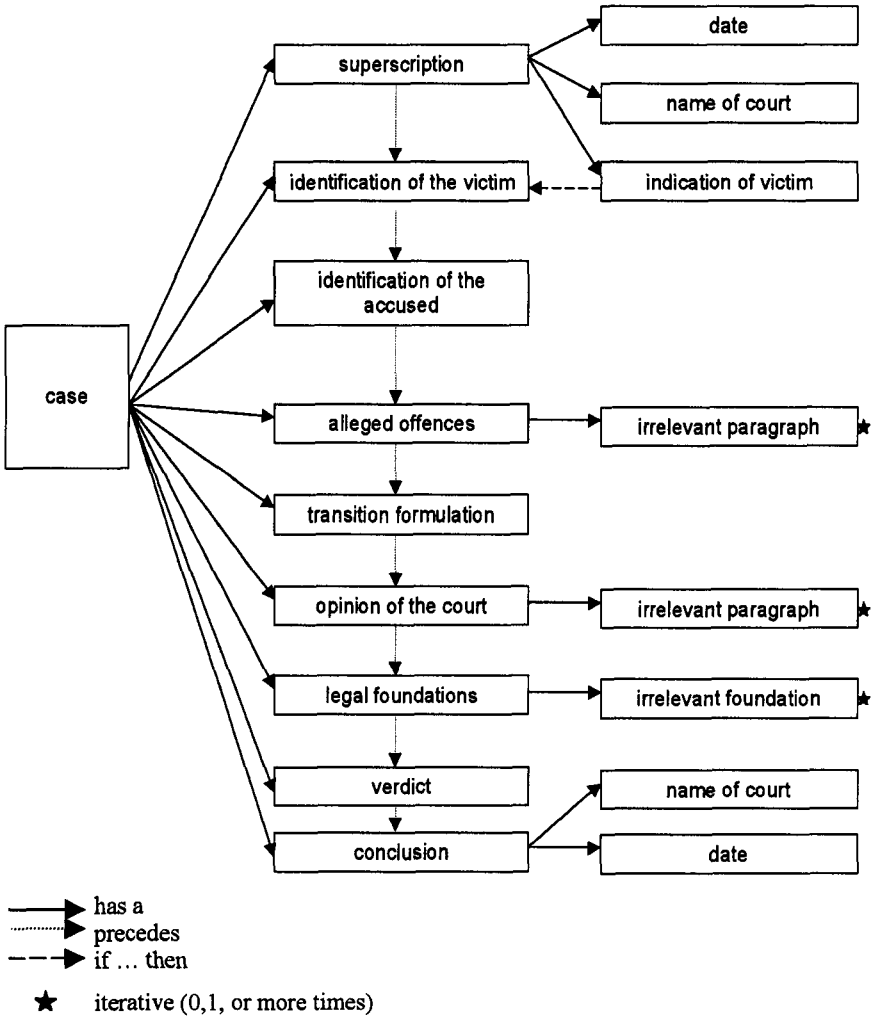


Figure 2. Example of a representation of the segments of a criminal case.

represents a typical text's sentence. A "phrase" segment is delimited by predefined punctuation marks (e.g., enumeration). The sentence as well as the phrase segment can be defined by delimiting or classifying word patterns. A "pattern" segment consists of a defined word pattern (e.g., template of a text string). A segment can have an interesting substructure. Then, the segment contains pointers to the subsegment frames. When the occurrence of a segment depends on other non-adjoining segment(s) of the text, a rule specifying this dependence is attached to the frame. Segments

have flags indicating whether they are optional or repetitive. When representing our criminal cases, we did not allow overlapping segments except in the case of nested segments.

The segment frames are organized as a semantic network (cf. Figure 2). The segment frames have a hierarchical (*has a*), sequential (*precedes*), or conditional relation (*if...then*) between them. The head segment frame defines the complete text or one major text component, and its possible subsegments. Such a representation is based upon a "top down" interpretation of the text: Its global concept is broken into more primitive concepts. Segments of a same hierarchical level may have a sequential relation: They follow one another in the text. The conditional relation is needed when the legitimacy of a segment depends upon the existence of another segment. The resulting scheme is an abstraction of the structure of the text as it is conceived by its class of users. It is possible to define different views (schemes) of the same text that each defines a different text use. Or, as it was the case for the criminal cases, to define different text categories, described by different text grammars and discriminated by different classifying word patterns of their head segment frames.

A text string or sequence of strings (word patterns, indicator words, or phrases) is an important indicator of the limits and/or category or class of a text segment. A segment can be characterized by a specific word pattern or by a logical combination of word patterns. Word patterns with a same delimiting or classifying function are grouped in a semantic class. A *word pattern frame* represents a semantic class and its member word patterns. This frame is connected with the appropriate text segment frame(s). A text segment frame has links to appropriate word pattern frames (*is_classified_by* or *is_delimited_by* relation).

Word patterns are regular expressions (expressions in regular grammar) and consist of one or more strings in a fixed order. Pattern elements are separated by spacing, or by punctuation marks and spacing. A pattern element is a word string, number, wild card, or word template. Wild cards represent random text and/or spacing. A word template is composed of fixed and wild card characters (e.g., the template "?laintiff?" representing "Plaintiff", "plaintiff", "plaintiffs", etc.). The wild cards of the templates allow for a selective normalization of text strings. In the representation of the criminal cases such templates were useful to represent dates, word stems, and the arbitrary use of capitals.

A delimiting or classifying word pattern may occur in the text in variant formulations. The variants are mainly lexical, morphological, and/or syntactical. It is important to control the number of word pattern variants (Figure 3) in the knowledge base. We could limit them by defining an

attribute in the pattern representation that allows facultative neglecting of punctuation marks, and by the use of wild cards as pattern elements or as string characters. The use of wild cards is very advantageous. The knowledge engineer defines the degree of fuzzy match between each word pattern and the text processed. More wild cards in a pattern increase the risk of an incorrect text interpretation by the system.

3.2 Parsing and Tagging of the Text

A parser was implemented to identify the category of the text and/or to recognize its components based upon the text grammar. A semantic network of frames represents the text grammar. Parsing a text based upon this network aims at recognizing nested segments, ordered segments, and segments the legitimacy of which depends upon the existence of other segments. The parser focuses on finding the segments defined in the text grammar, while neglecting the remainder of the text. The parser can be considered as a "*partial parser*", which targets specific information, while skipping over other text parts.

The nested structure of segments (*has a* relation) is described by a context-free grammar, represented by a tree structure. The parsing starts with the triggering of the head segment frame. When a segment is identified, its subsegments (siblings) will be searched. A sibling inherits from its parents the text positions between which it is to be found. The segment tree is accessed with a depth-first strategy: Subsegments are identified before other segments on a same hierarchical level are searched. The parsing employs a push down stack in order to remember segment frames still to be processed (cf. a *push down automaton*). Segments of the same hierarchic level, possibly but not necessarily follow each other in the text. The recognition of segments on a same hierarchical level takes into account the *precedes* relation, when defined in the grammar. The activation of a frame may depend upon the existence of a specific text segment (*if ... then* relation) already found in the text. In this case the frame is activated after positive evaluation of the production rule attached to the frame.

The recognition of a segment takes into account its type and its *classifying* and *delimiting* patterns. When categorization of a segment depends upon a word pattern or a logical combination of word patterns, the parser employs a separate module, a *finite state automaton*, which recognizes regular expressions in the text in an efficient way. A fuzzy search or probabilistic ranking of the match between the word pattern and the text is not applied. The knowledge engineer himself defines locations in the pattern where an inexact match is approved.

Given the documents of the preliminary inquiry;
Given the documents of the judicial inquiry;
The court has examined
The Court has examined
Since the plaintiff does not master the Dutch language
Since the plaintiffs do not master the Dutch language
Given ... grounds

Figure 3. Example of some pattern variants of the semantic class "begin transition".

The parser is *deterministic*: Alternative solutions are ordered by priority. Most text characteristics uniquely define the text segments. A backtracking mechanism would not necessarily result in a better parsing. When a text is processed it is important to detect an ungrammatical situation at the place of occurrence and not interpret this situation as the result of an incorrect previous decision. So, the ungrammatical situation can be optimally corrected for further parsing (Charniak, 1983). For instance, when a segment is not optional and only one of the segment limits is positively identified, the whole segment can be identified at this limit, thus minimally disturbing the processing of other segments.

After a segment is found, its begin and end positions in the text are marked with the segment name. Tags in *SGML*-syntax are attributed. Except for the attribution of category tags, the parsing does not structurally, lexically, morphologically, or syntactically alter the original text. Figure 4 shows an example of a tagged criminal case.

4. RESULTS AND DISCUSSION

The SALOMON system was applied upon Belgian criminal cases issued by the correctional court of Leuven, dating from 1992-1994. The system realizes an essential categorization of the criminal cases. Also the structuring of the criminal case in relevant and irrelevant segments and subsegments is accomplished.

The text grammar knowledge related to the 23 categories, the ca. 300 word patterns (consisting of an average of 3.5 strings, numbers, or templates) organized in 31 classes, and the more than 100 relations between text segments was acquired and implemented in respectively 11 and 5 man days. Some necessary corrections of and additions to the knowledge base, carried out after processing and evaluating an initial sample of 25 cases, required 3 man-days.

<appeal_procedure>
<superscription> *Court Administration number: ...*
<court> *Correctional Court Leuven* </court> ...
<date> *January 20,1993* </date> ...
In the case of the Public Prosecutor and of:
</superscription>
<victim> ...
</victim>
<accused> *Against ...*
Defendant in opposition ...
</accused>
<alleged_offences>
<irrelevant_paragraph_alleged_offences> *..Accused: ...*
</irrelevant_paragraph_alleged_offences> ...
<irrelevant_paragraph_alleged_offences> *... By reason of...*
</irrelevant_paragraph_alleged_offences>...
</alleged_offences>
<transition_formulation> *Given the documents in the case ...*
Given the Public Prosecutor's case for the prosecution
</transition_formulation>
< opinion_of_the_court> *Whereas ...*
<irrelevant_paragraph_opinion> *...offence ... is certain...*
</irrelevant_paragraph_opinion> ...
<irrelevant_paragraph_opinion> *Given the enactment...*
</irrelevant_paragraph_opinion> ...
</opinion_of_the_court>
<legal_foundations> *On these grounds and in application of the following statutory provisions ...*
<irrelevant_foundations> *... Code of criminal procedure...*
</irrelevant_foundations>
</legal_foundations>
<verdict> *THE COURT ...*
</verdict>
<conclusion> *Thus given ...*
</conclusion>
</appeal_procedure>

Figure 4. Example of a SGML-tagged case. The word patterns in italic classify or delimit the case or its segments.

The result of the parsing of a criminal case is a case text indicating the general category (Figure 4). General decisions are distinguished from the special ones (decisions about appeal procedures, civil interests, refusals to witness, false translations by interpreters, infringements by foreigners, and internment of people). Moreover, the case segments defined in the text grammar are identified and tagged including the superscription, identification of the victim, identification of the accused, alleged offences, transition formulation, opinion of the court, legal foundations, verdict, conclusion, date, name of the court, irrelevant paragraphs of the alleged offences and of the opinion of the court, and irrelevant foundations (Figure 4). From the tagged case general information about the case such as the date, the name of the court, and relevant legal foundations are easily extracted and placed in the case abstract. The remaining relevant parts of the alleged offences and opinion of the court are ready for further abstracting with shallow statistical techniques (see chapter 8).

A sample of 1000 criminal cases (test base) was drawn from the original corpus. This test set is distinguished from the case set employed for knowledge acquisition. It is composed of 882 general and 118 special decisions, a proportion representative for the complete corpus. We use the definitions of *recall*, *precision*, *fallout* and *error rate* that are commonly applied in text categorization (see chapter 5, p. 104). The metrics compare the output of an automatic categorization with the manual categorization by an expert. Our expert was not a member of the research team, but an outsider, namely a student entering her final year of law school. The expert intellectually categorized the test criminal cases and their segments. She also intellectually marked paragraphs of the offences charged and of the opinion of the court irrelevant for inclusion in the case summary. The results were compared with the output of the SALOMON system.

Recall and *precision* are calculated for all categories (Tables 1 and 2). We computed *fallout* and *error rate* for segments with fixed limits (e.g., entire case, "paragraph" and "phrase" segments) (Tables 1 and 3). For case segments, we separated the results of the processing of general and special decisions. In this way the types of errors are illustrated. For the case category an average recall and precision of respectively 95% and 99% are achieved. For the case segments average recall and precision values of respectively 88% and 93% for general decisions and respectively 66% and 88% for special decisions are obtained. In general, precision is higher than recall. Recall errors are usually the result of lack of knowledge such as missing relations or word patterns (e.g., a zero recall of the category "name of court conclusion" for special decisions), whereas precision errors are often due to ambiguities in the knowledge. Typing errors cause a substantial number of

errors. For instance, in the category "date superscription" 90% and 57% of the errors for respectively general decisions and special decisions responsible for the non-identification of this category regard spelling errors (e.g., no space between the date and a foregoing word). For instance, in the category "irrelevant foundations'' 88% and 83% of the errors for respectively general decisions and special decisions responsible for the non-correct identification of this category regard the improper use of punctuation marks (e.g., no space between the punctuation mark and the following word). The non-identification of a parent segment sometimes explains a low recall of its subsegment (e.g., the categories "date conclusion" and "irrelevant foundations" for special decisions). The use of wild cards in the representation of the patterns did not cause any misinterpretation by the system. The overall results are satisfying taking into account the limited time for knowledge acquisition and implementation.

As a consequence of the structuring of the case, *routine cases* are identified. The opinion of the court part of a routine case consists entirely of routine or irrelevant text. Table 4 summarizes the evaluation of the assignment of cases to the class of routine cases. The routine cases are recognized with an error rate of 14%. A recall of 63%, a precision of 99%, and fallout of near to 0% signify that the system does not find all routine cases, but the ones that are identified are correctly recognized. The relatively low recall value is explained by the limited amount of word patterns used for recognizing irrelevant text in the opinion of the court. A larger number of patterns would increase the risk of subjective interpretations by the knowledge engineer who selects the patterns (Uyttendaele et al., 1998). Routine cases are of no relevance for the legal professional. The system correctly discards almost two thirds of the routine cases.

5. CONTRIBUTIONS OF THE RESEARCH

In the chapters 5 and 6 we saw that a manually constructed knowledge base for a particular subject area is not uncommon in automatic text analysis. Knowledge bases have been successfully used in *text categorization*, *information extraction from text*, and *text summarization*. Successful systems heavily rely upon knowledge of the subject domain. The systems often parse the texts in order to identify domain concepts or their variant forms.

When texts cover unrestricted subject areas, it is equally useful to identify where in the text significant information is to be found. The legal cases discuss a large variety of crime topics and facts. Human readers can reliably identify relevant texts or relevant portions of texts merely by

skimming the texts for cues. *Cue words*, indicator phrases, and context patterns have been employed to identify significant sentences and concepts in texts for abstracting and classifying purposes (see chapter 6). When skimming a text, knowledge of the *text structure* of the text type is also advantageous. Text structure refers to the organization and interconnections between textual units, such that text conveys a meaningful message to the reader. Automatically simulating a first rough skim of a document text, while employing knowledge about text cues and structure, has multiple application potentials including automatic categorization, indexing and abstracting.

As seen in chapter 2, the discipline of text linguistics considers the complete text a superior grammatical unit. In the same way as the form of sentences is described in terms of word order (syntax), we can decompose the form of whole texts into a number of fixed, conventional components or categories and formulate rules for their characteristic order. This leads to the idea of representing the structural aspects of a text by means of a text grammar. There is a choice of forms to represent a text grammar. When organized in a semantic network, *frames* are well suited to represent the document structure. In addition, they easily represent a hierarchy of topics and subtopics. Text grammar research is still in its infancy. Our research is a modest attempt to use a text grammar in text analysis.

We have implemented a *domain-independent formalism*, which allows the representation of text structure including the major semantic units of a text, their attributes, and relations in the form of a text specific grammar. The formalism also allows representing the concepts typical of the criminal law domain. In addition, it can represent different views of a text, each defining different uses. Parsing of the criminal cases based upon a case specific grammar results in categorization of the cases, identification and categorization of relevant case components, and identification of insignificant case components. This procedure is a first, important step towards automatically abstracting legal cases.

The research demonstrates that the *discourse structures* of the legal case and their *surface linguistic phenomena* are useful in automatic abstracting. Knowledge of the superstructure of the case, which forms the overall organization structure of the information, is beneficial to recognize the main components of a case. Within these components some specific topics or information can be located. The legal field has specific text types. This makes discourse structure especially useful in text analysis. A second interesting aspect of discourse analysis involves the study of the surface linguistic phenomena that depend on the structural aspects of discourse. Creators of texts often use specific linguistic signals (e.g., cue words and

phrases) that indicate the category of the text constituent, topic shifts, or changes in an argumentation structure. Legal texts are often formulaic in nature and thus provide excellent cue phrases. However, we need more discourse analytical studies that identify the rules and conventions that govern the discourse of a number of legal text types. Such studies yield valuable knowledge to be incorporated in text analysis and perhaps in text generation (Moens, Uyttendaele, & Dumortier, 1999b).

Table 1. Results of the categorization of the entire criminal case.

Case category	Effectiveness measures			
	Recall	Precision	Fallout	Error rate
Appeal procedures	1.000000	1.000000	0.000000	0.000000
Civil interests	1.000000	0.916667	0.001011	0.001000
Refusals to witness	0.888889	1.000000	0.000000	0.001000
False translations	1.000000	1.000000	0.000000	0.000000
Infringements by foreigners	0.733333	1.000000	0.000000	0.004000
Internment of people	1.000000	1.000000	0.000000	0.000000
General case	1.000000	0.994363	0.042373	0.005000
Average	0.946032	0.987290	0.006198	0.001571

Table 2. Results of the categorization of the case segments. Note: -- = not defined (the category does not apply or division by zero).

Case segment category	Effectiveness measures			
	General decisions		Special decisions	
	Recall	Precision	Recall	Precision
Superscription	0.970522	0.970522	0.771186	0.784483
Date superscription	0.916100	0.987775	0.866667	0.939759
Name of court superscription	0.987528	0.996568	0.814159	1.000000
Identification of the victim	0.743935	0.862500	0.575000	0.920000
Identification of the accused	0.787982	0.794286	0.745763	0.846154
Alleged offences	0.843964	0.982759	0.696629	0.925373
Irrelevant paragraph offences	0.819536	0.966945	0.812155	0.954545
Transition formulation	0.867347	0.891608	0.500000	0.632184
Opinion of the court	0.871882	0.895227	0.594595	0.687500
Irrelevant paragraph opinion	0.856416	0.991582	0.907143	0.980695
Legal foundations	0.910431	0.931555	0.813084	0.861386
Irrelevant foundations	0.769907	0.793555	0.688679	0.768421
Verdict	0.896825	0.933884	0.703390	0.954023
Conclusion	0.959184	0.998819	0.728814	1.000000
Date conclusion	--	--	0.375000	1.000000
Name of court conclusion	--	--	0.000000	--
Average	0.87154	0.928399	0.662017	0.883635

Table 3. Fallout and error rate of the categorization of the segments with fixed limits.

Case segment category	Effectiveness measures			
	General decisions		Special decisions	
	Fallout	Error rate	Fallout	Error rate
Irrelevant paragraph offences	0.026942	0.102202	0.030882	0.100572
Irrelevant paragraph opinion	0.006897	0.073438	0.010267	0.040417
Irrelevant foundations	0.099805	0.143136	0.173228	0.236052

Table 4. Results of the categorization of the routine cases.

Case category	Effectiveness measures			
	Recall	Precision	Fallout	Error rate
Routine case	0.625000	0.993976	0.002294	0.142857

Supervised learning techniques are unmistakably advantageous for the acquisition of simple lexical-semantic patterns that classify texts (see chapters 5 and 10). In the above application, the automatic learning of the complex text grammar that classifies a criminal case and its segments did not seem beneficial. Apart from the difficulty of learning the complete text structure from example texts, including all relevant and irrelevant text segments and their relations, there are the complications in automatically acquiring the word patterns that delimit or classify texts. Complex patterns (combinations in propositional logic of simple patterns) classify the texts of the criminal cases. The more "simple" patterns are not restricted to a specific type. They could be single words, phrases, and consecutive words with no syntactic relation, or whole sentences. Apart from the reasonable chance of an incorrect learning of the patterns, it was found that at least an almost similar effort would be needed to sample enough representative examples and carry out the manual tagging of the categories in these examples, as the effort needed for manually constructing the knowledge base. However, the machine learning techniques could be useful for learning specific word patterns. In the current application, it is not always possible to recognize all routine cases because of a lack of objective text patterns. An expert can easily identify a routine case. Determining the exact words and phrases that convey a routine sentence in the opinion of the court is much more difficult. Here, the machine might perform a more objective job, when learning the patterns from examples that are tagged as routine.

6. CONCLUSIONS

An initial text categorization and structuring is useful for many purposes of text analysis including automatic abstracting. The recognition of the text category, and of relevant and insignificant text components is an important first step when intellectually abstracting. Automating this process was especially useful for controlling the overload of present and future court decisions.

Knowledge of the discourse structures of criminal cases proved to be very helpful in automatically extracting relevant information from the cases and in automatically abstracting them. The patterns of the discourse involved in an automatic categorization and structuring of the legal cases are complex, but the number of patterns is limited, hence our choice of a manually constructed knowledge base for analyzing the cases. A powerful formalism for representing the knowledge is needed. It has been shown that a representation as a text grammar is very promising. Our research is a step towards generic representations of knowledge about discourse patterns.

However, in spite of the potential of the knowledge of discourse patterns in text analysis, intertextual analysis of the constitution of texts in terms of types and genres is underdeveloped in the legal field.

Chapter 8

CLUSTERING OF PARAGRAPHS WHEN SUMMARIZING LEGAL CASES

1. INTRODUCTION

As it is described in the previous chapter, the SALOMON project developed and tested several techniques to automatically summarize Belgian criminal cases.

In a first step, the case category, the major semantic case components, some general information (e.g., date, court name, relevant legal foundations) and the non-relevant paragraphs of the text are identified using a text grammar approach. In a second step, relevant paragraphs of the text of the offences charged and of the opinion of the court are further abstracted (see Chapter 7, Figure 1). Because their detailed content is unpredictable and relates to a broad subject domain, the thematic structure, key paragraphs, and key terms of these texts are identified with *shallow statistical techniques*. This second step is the main subject of this chapter (see also Moens, Uyttendaele, & Dumortier, 1997, 1999a). The first step was the subject of the previous chapter.

This chapter is organized as follows. We describe the characteristics of the offences and opinion texts of the cases. The methods are discussed in detail and are followed by an evaluation of the results. We demonstrate that clustering algorithms based on the selection of representative objects have a definite potential for automatic abstracting as well as for the recognition of the thematic structure of text. We finish with summing up the contributions of the research.

2. TEXT CORPUS AND OUTPUT OF THE SYSTEM

The SALOMON techniques were developed in order to extract and summarize the most relevant case components: alleged offences, opinion of the court, and legal foundations. The abstracting of the legal foundations is discussed in the previous chapter. This chapter discusses the abstracting of the alleged offences and the opinion of the court. The texts of the offences charged and the opinion of the court are often long and elaborated. They are characterized by especially long sentences of an average length of 3 to 5 text lines. Because a new line character separates the sentences, we call them paragraphs. The text of the opinion of the court is of varying size. It can contain 50 paragraphs or more. The size of the offence text seldom exceeds 15 paragraphs. These sizes do not include routine paragraphs that were eliminated during the first abstracting step. The texts of the offences and opinion of the court deal with any criminal aspect of society.

The *alleged offences* describe the crimes a person is accused of. Each crime or delict is described in a separate paragraph called a "delict description". A delict description also contains the specific facts and circumstances of the delict, which are integrated in the text that describes the crime. Elaborate offences contain a bulk of redundant material. Some of the delict descriptions describe the same crime, but refer to different facts or accused. The crime concepts mentioned in the delict descriptions are usually disclosed in a fixed, stereotypical way. We want to eliminate redundant delict descriptions and to extract key paragraphs.

The *opinion of the court* contains the argumentation of the judge regarding the crimes committed. The opinion of the court often discusses different themes or topics. A theme may be abandoned and resumed during the discourse. The text contains routine, factual, and principle paragraphs. Routine grounds are standard formulations that are of no relevance. They were identified in the first abstracting step. Factual grounds are the considerations of the judge regarding the facts of the crime. In the principle grounds, the court gives general, abstract information about statute application and interpretation. The *opinion of the court* allows distinguishing three types of cases within the studied corpus: *routine cases* (containing only routine, unimportant grounds in their opinion), *non-routine cases* (containing other than routine-grounds) and *leading cases* (containing more than 5 principle grounds). The leading cases only represent 3 to 5% of the total corpus. We want to identify key paragraphs, which possibly represent principle grounds, and representative key terms in the opinion of the court.

3. METHODS: THE CLUSTERING TECHNIQUES

We employ *non-hierarchical clustering methods* that are *based on the selection of representative objects* to thematically group the paragraphs of alleged offences and opinion of the court, and to identify representative paragraphs. The representative paragraphs are extracted. They form the summary of the alleged offences and opinion of the court.

Cluster analysis is a multivariate statistical technique that automatically generates groups in data. It is considered as unsupervised learning. Non-hierarchical methods partition a set of objects into clusters of similar objects. *Clustering* methods *based on the selection of representative objects* consider possible choices of representative objects and then construct clusters around them. The technique of clustering supposes:

1. an abstract representation of the textual object to be clustered, containing the text features or attributes for the classification;
2. a function that computes the relative importance (weight) of the features; a function that computes a numerical similarity between the representations.

Each paragraph of the text of the alleged offences and opinion of the court is represented as a term vector. The terms (single words) are selected after elimination of stopwords and proper names, and are currently not stemmed. Stopwords are identified as the most frequent words in the corpus of legal cases. Proper names are recognized as capitalized words. The terms of the alleged offences are weighted with the *in-paragraph frequency*, which is computed as the number of times a term *i* occurs in the text paragraph. Considering the stereotypical way used in describing the crimes committed, less important content words also contribute to identifying redundancy. Discriminating the terms of the opinion of the court is done with *inverse document frequency weights*, which are computed before the actual abstracting. Their computation is based upon about 3000 cases and results in a list of term weights. Numbers are not included in the term vectors of the opinion paragraphs. The similarity between two text paragraphs is calculated as the cosine coefficient of their term vectors representations *V1* and *V2* (cf. Jones & Furnas, 1987):

$$\frac{\sum_{i=1}^{n} V1_i \cdot V2_i}{\sqrt{\sum_{i=1}^{n} V1_i^2 \cdot \sum_{i=1}^{n} V2_i^2}} \qquad (1)$$

where

n = number of distinct terms in the paragraphs to be clustered.

In preliminary experiments, the cosine function performed better than the inner product as similarity coefficient because of length normalization.

Clustering methods based on the selection of representative objects consider possible choices of representative objects (also called *centrotypes* or *medoids*) and then construct clusters around them. We adapted and further developed clustering algorithms described by Kaufman and Rousseeuw (1990, p. 68 ff.) for use in text-based systems. In the algorithms employed, each object can only belong to one cluster. As in other non-hierarchical methods, these algorithms split a data set of *n* objects into *k* clusters.

We implemented the *covering clustering algorithm* for clustering of identical delict description paragraphs of the alleged offences disturbed by different facts, or with a variant sentence structure, and to eliminate redundant delict descriptions (Figure 1). In this algorithm, possible representative paragraphs (medoids) are considered for a potential grouping, but each paragraph must at least have a given similarity (threshold) with the representative paragraph of its cluster. The objective is to minimize the number of representative paragraphs. The threshold value is useful to define the degree of redundancy allowed and was set after several trials. We added an extra constraint: For a given number of medoids, a best solution is found for which the total (or average) similarity between each non-selected object (paragraph) and its medoid is maximized. We implemented a best solution to this problem with the following algorithm, which considers n! / (k! (n - k)!) possible solutions for each value of *k*. The number of *k*-values to be tested depends upon how fast an acceptable solution is found.

```
Covering Algorithm

define threshold
init k = 1
WHILE (k <= n ) AND not found acceptable combination
  FOR each possible combination of k medoids (= selected objects)
      FOR  each non-selected object
           determine its medoid
      IF combination of medoids is acceptable (= each non-selected object
      has a similarity above the threshold with the medoid of its cluster)
      THEN calculate total similarity of each non-selected object and its medoid
  IF an acceptable combination is found
  THEN select acceptable combination of k medoids for which the total similarity
  of each non-selected object and its medoid is maximized and the algorithm stops
  ELSE increase k with 1
```

We implemented the *k-medoid method* for clustering the paragraphs of the opinion of the court according to theme (Figure 2). The k-medoid method searches the best possible clustering in *k*-groups of a set of objects. The optimal solution of this problem is the generation of all possible *k* representative paragraphs (medoids) and the choice of the best possible solution for which the total (or average) similarity of each non-selected object (paragraph) and its medoid is maximized. We implemented a best solution to this problem with the following algorithm.

k-Medoid Method: Best Solution

```
define k
FOR each possible combination of k medoids (= selected objects)
   FOR  each non-selected object
          determine its medoid
   calculate total similarity of each non-selected object and its medoid
select combination of k medoids for which the total similarity of each non-selected object
and its medoid is maximized
```

An optimal solution, which for the chosen *k* value considers n! / (k! (n - k)!) possible combinations, is only executable for relatively small problems. We implemented an optimal solution for up to 15 paragraphs to be clustered. Because the texts of the opinion of the court may contain more than 50 paragraphs, we implemented a good, but not optimal solution for the *k*-medoid method. The algorithm can be considered as a reallocation algorithm. An initial clustering is improved in consequent steps until a specific criterion is met. The algorithm consists of two phases. First, an initial clustering is performed by successive selection of representative paragraphs (medoids) until *k* medoids are found (function BUILD). Second, to improve the clustering yielded by BUILD, the set of all pairs of objects (*i*,*h*), for which object *i* has been selected as representative paragraph (medoid) and object *h* has not, will be considered in the search for a better clustering (function SWAP).

As an initial step, the function BUILD selects the most centrally located object of the data set. The object is chosen for which the sum of similarities to all other objects is maximized. This object is the first medoid. In the next steps, each time a new medoid is chosen until *k* medoids are found. The medoid chosen is the object for which a maximum gain in total (or average) similarities between each non-selected object and its medoid is obtained.

For a given initial clustering, the function SWAP considers each pair of objects (*i*,*h*) for which *i* has been selected as representative object and *h* not.

For each pair, the contribution to the clustering is computed when representative object *i* is replaced by object *h*. This contribution is positive (increase in total or average similarity values between each non-selected object and its medoid), negative (decrease in total or average similarity values between each non-selected object and its medoid), or zero. The swap-pair with the highest contribution is selected. If this contribution is positive, the swapping operation is executed, and the whole procedure of calculating the contribution of all possible swapping operations is repeated, otherwise the algorithm stops (no better grouping can be found).

This reallocation algorithm is computationally much less expensive than a best solution to the problem. For long texts, usually a few swapping operations are sufficient to obtain a good clustering.

k-Medoid Method: Good Solution

define *k*
1) BUILD
select most centrally located object of the data set (sum of similarities with all other objects is maximized)
build a cluster around this medoid
REPEAT
  FOR each candidate medoid *i*
    FOR each non-selected object *j*
      calculate similarity between *i* and *j* = s(*i*,*j*)
      compare s (*i*,*j*) with S_j (similarity between *j* and the medoid of the cluster *j* currently belongs to)
      IF S_j <s(*i*,*j*)
      THEN compute the gain in similarity when moving *j*
    compute the total gain in similarities by choosing object *i* (L_i)
  choose the best *i* for which L_i is maximized
  build clusters around the medoids
UNTIL *k* medoids are found
2) SWAP
FOR each pair (*i*,*h*) (*i* is selected *h* not)
    calculate CONTRIBUTION TO THE CLUSTERING
select pair for which contribution to the clustering is maximized
IF contribution is positive
THEN execute swapping operation with selected pair
      repeat SWAP
ELSE stop SWAP

The contribution of swapping the pair (*i*,*h*) is computed as the total of changes in similarities, when *h* becomes medoid instead of *i*. Instead of recalculating the total similarities in this new cluster structure, only those similarities that are affected by the change in cluster structure are computed.

CONTRIBUTION TO THE CLUSTERING

The changes regarding *i* and *h*:
i becomes a member instead of a medoid: its new medoid is searched and the similarity between *i* and this medoid is added to the contribution
h becomes a medoid instead of a member: its old medoid is searched and the similarity between *h* and this medoid is subtracted from the contribution

The changes regarding all other non-selected objects *j* are added to the contribution:
IF *j* is more similar to one of the other medoids than to *i* or to *h*
THEN *j* does not change position in the cluster structure and the contribution = zero
ELSE
IF *i* was the medoid of the cluster *j* belongs to
THEN IF *j* is closer to *h* than to its second choice medoid (*x*):
THEN *j* changes from. cluster with medoid *i* to cluster with medoid *h*: the contribution is positive, negative, or zero, depending on the difference in similarities between *j* and *h* and *j* and *i* (sim (*j*, *h*) - sim(*j*, *i*))
ELSE *j* changes from cluster with medoid *i* to cluster with medoid *x* (*x* = second choice medoid of *j*): the contribution is negative or zero, depending on the difference in similarity between *j* and *x* and *j* and *i* (sim (*j*, *x*) - sim(*j*, *i*))
ELSE IF the similarity between *j* and *h* is higher than the similarity between *j* and its current medoid *y*
THEN *j* changes from cluster with medoid *y* to cluster with medoid *h*: the contribution is always positive and represents the difference in similarity between *j* and *h* and *j* and *y* (sim(*j*, *h*) - sim(*j*, *y*))

As all combinations of medoids (in case of an optimal solution) or all potential swapping operations are considered (in case of a good solution), the results of the algorithms do not depend on the order of the objects in the input file (except in case the similarities between objects are tied).

The number of medoids (*k*) is predefined or is determined as part of the clustering method. In the latter case, employed in SALOMON, possible *k* values are considered in the search for the best *k* value. For each object *i* of the cluster structure, the degree of fitness (*f(i)*) of an object *i* to its cluster is computed as the normalized difference between the average similarity of the

object *i* to all other objects of its cluster and the similarity of *i* with its second choice cluster:

$$f(i) = (a(i) - b(i)) / \max (a(i), b(i)) \quad (2)$$

where:

$a(i)$ = average similarity of *i* to all other objects of its cluster

$b(i)$ = maximum of the similarities of *i* with each other cluster whereto *i* does not belong computed as the average similarity of *i* with the objects of this cluster, i.e., the similarity of *i* to its second choice cluster.

PARAGRAPHS OF THE ALLEGED OFFENCES =

import: namely cannabis from The Netherlands (Maastricht) (O.S. 91/2068);

In breach of article 1,2 b (1 and 5) of the Act of 24 February 1921, and of article 1, 3, 11 and 28 of the Royal Decree of 31 December 1930 on Drugs and Narcotics, having imported, possessed, sold or offered for sale narcotics or other psychotropic drugs that may induce dependence and that are enlisted by Royal Decree, for valuable consideration or for free, without preceding license of the Ministry of Public Health, namely ...

possession: several times cannabis, as it turns out from the analysis of exhibits O. S. 90/1571 and 91/2068

possession: several times cannabis

REPRESENTATIVE PARAGRAPHS =

In breach of article 1,2 b (1 and 5) of the Act of 24 February 1921, and of article 1, 3, 11 and 28 of the Royal Decree of 31 December 1930 on Drugs and Narcotics, having imported, possessed, sold or offered for sale narcotics or other psychotropic drugs that may induce dependence and that are enlisted by Royal Decree, for valuable consideration or for free, without preceding license of the Ministry of Public Health, namely ...

possession: several times cannabis

Figure 1. Brief example of the elimination of redundant paragraphs in the alleged offences (translated from Dutch).[1]

For each possible k value (except for $k = 1$ or $k = n$), we compute a best or good clustering, compute the degree of fitness of each object to its cluster, and average these fitness values. The best k value is the one for which the average fitness value is maximized. To test whether $k = 1$ (in case the best $k = 2$) or $k = n$ (in case the best $k = n - 1$) represents a better clustering, we respectively test whether the average similarity between each non-selected object and its medoid increases or whether the average similarity between objects of different clusters decreases. For the former test, we first compute the medoid when $k = 1$.

The *medoid of each cluster* or most centrally located object of the cluster forms a representative description of each crime or topic treated in the alleged offences or opinion of the court (Figure 2). We assume that a text sentence or paragraph that is closely linked by patterns of content words to a number of other text sentences or paragraphs is informative, and thus is relevant to include in the summary (cf. Prikhod'ko & Skorokhod'ko, 1982).

In addition to the paragraphs, we also extract *key terms* from clusters of opinion of the court paragraphs that contain more than three objects (Figures 3 and 4). Different methods are possible for key term selection (Jardine & van Rijsbergen, 1971; Willett, 1980). Currently, we select the two terms with highest weight from the terms of the average vector of the cluster.

Presently, we limit ourselves to the extraction of information from the case text. No attempt is made to re-edit this information. Given the danger of misinterpreting or misrepresenting the case text, even abstracts of legal cases that are intellectually composed are no more than the extraction of relevant text parts (Uyttendaele et al., 1996, 1998).

4. RESULTS AND DISCUSSION

The recognition of representative paragraphs of the alleged offences and opinion of the court is evaluated upon 700 criminal cases. The test set is representative for the complete corpus. Evaluation of text abstracts is a difficult task. An intuitive approach is to compare the abstract automatically generated with the abstract intellectually produced by an expert. Our expert was not a member of the research team, but an outsider, namely a student entering her final year of law school. The expert intellectually marked paragraphs of the offences charged and of the opinion of the court relevant for inclusion in the case summary. The results were compared with the output of the SALOMON system. For each criminal case, the expert determined the crime themes that were discussed in the *alleged offences*. Knowledge of the law, which defines the crime themes, facilitates this job.

Then, the expert associated each paragraph of the alleged offences with the correct crime theme. For each crime theme group, she determined the correct representative paragraph as the paragraph that conveyed the most information about the crime concept, i.e., the most complete crime description when compared to the law. She also determined partially correct representative paragraphs as the ones that are not figured in the law as such, but that are still informative enough about the crime concept. This procedure was repeated for the *opinion of the court* part of the case text. However, this manual task is much more subjective, especially when the expert has to select the paragraph from a group of paragraphs that is most informative of the content of the topic cluster. We realized that the identification of paragraphs in the opinion of the court that reflect the topics of the argumentation of the judge is sometimes a subjective operation and is ideally repeated by different experts. Due to a limited timing and tight financial circumstances, it was not possible to have the evaluation of the paragraphs of the opinion of the court repeated by other experts, nor to have the extracted key terms evaluated.

We use the metrics recall, precision, overgeneration, and fallout as applied in the field of *text extraction* (see chapter 6, p. 135). These metrics compare representative paragraphs intellectually identified with paragraphs automatically generated. We assign a weight of 0.5 to partial correct responses. The metrics are calculated for each text of the offences and the opinion of the court of 700 criminal cases of the test set and are averaged.

A "*methodological*" *evaluation* (Table 1) aims at evaluating extracted paragraphs in representing the topics of the abstracted text. The high *recall* (97% and 85% for, respectively, alleged offences and opinion of the court) and precision (95% and 81% for, respectively, alleged offences and opinion of the court) values, and the restricted *fallout* values (28% and 24% for, respectively, alleged offences and opinion of the court) are satisfying. They indicate that the techniques employed are suitable for recognizing the theme structure of the legal texts and for identifying representative text paragraphs. *Overgeneration* of responses is low (4% and 9% for, respectively, alleged offences and opinion of the court), indicating that the system rather robustly identifies the number of themes in the text. The main errors are due to morphological variants of related and identical concepts, and to incorrect orthographic boundaries in the original text. The standardized naming and description of legal concepts in the offences make a thematic grouping and recognition of redundant material very effective, and explain the better results of structuring the alleged offences.

EXAMPLE OF A CLUSTER OF PARAGRAPHS =

Whereas the accused keep on referring wrongfully to folklore and tradition when it comes to cockfighting.

Whereas it is moreover very shocking to be a supporter of cockfighting and to fully approve of it, while in the meantime not have the courage either to be open about one's view, or to accept responsibility for acts committed in accordance with one's conviction.

That the defense bases its case wrongfully on the fact that elsewhere, that is outside Belgium, no legal action is taken against cockfighting and or that there exist numerous other practices of animal abuse.

Whereas the accused hide themselves wrongfully behind the alleged nature of the animal; it is up to man, as a being with the power of reason and emotion, to protect animals against this and not have them suffer needlessly from their instincts; not to mention the fact that "nature" is exploited to satisfy various base sensations and these just have to give way to higher feelings of protection and responsibility towards animals who need man in any case.

Whereas every available means should be used for this purpose, including amongst others, the prohibition to keep animals as provided by the law, since this is a useful and necessary means to urge the accused to reflect on and respect the animal.

REPRESENTATIVE PARAGRAPH =

That the defense bases its case wrongfully on the fact that elsewhere, that is outside Belgium, no legal action is taken against cockfighting and or that there exist numerous other practices of animal abuse.

Figure 2. Example of a cluster of paragraphs and its representative paragraph in the opinion of the court (translated from Dutch).

Table 1. Average results of a ''methodological'' evaluation of the abstracting of alleged offences and opinion of the court.

	Effectiveness measures			
	Recall	Precision	Overgeneration	Fallout
Alleged offences	0.972664	0.954161	0.037021	0.282887
Opinion of the court	0.847502	0.810143	0.085496	0.239101

The non-hierarchical clustering algorithms based on the selection of representative objects do not rely upon an average cluster representative, which is dynamically computed as the cluster structure is built. So, the clustering is not dependent upon the order of input (except in the case of ties). The algorithms select, for each cluster, a cluster representative that is a natural object of the cluster. A representative sentence or paragraph in a cluster of related sentences or paragraphs is especially valuable for abstracting purposes. When abstracting the opinion of the court, a clustering algorithm that does not rely on threshold values is employed. In this way, we obtain a natural clustering, necessary to generate a balanced summary. The algorithms are computationally expensive, but they satisfy a present interest in methods that employ the power of current computers to search and iterate until they achieve a good fit to the data. Processing time is acceptable (on average, about 1.5 seconds per case text and a few seconds for a long text on a Sun™ SPARC station 5 (85 MHz)).

The clustering algorithms have been productive in theme and text structure recognition of our legal texts. The algorithms have a potential for automatic abstracting, indexing, and text linking. However, we think that much of the success of the shallow statistical techniques is due to the fact that the texts exhibit a very *stereotyping naming* of the terms that describe the crime concepts, which are the concepts to include in the case summary. On the other hand, concepts that are not relevant to include in the summary (e.g., description of the circumstances of the crime) exhibit much more variety in their word use. This situation facilitates a thematic grouping around crime concepts with the help of shallow statistical techniques.

Summarizing the alleged offences and opinion of the court is part of the larger process of abstracting legal cases (see chapter 7; Moens & Uyttendaele, 1997; Moens, Uyttendaele, & Dumortier, 1997). A "*legal*" *evaluation* (Table 2) judges the final quality of abstracted offences and opinion of the court for the legal professional and aims at detecting the limits of our approach. Here relevancy relates to the identification of distinct delict descriptions (in the offences) and to the value in indicating legal principles (in the opinion). This evaluation takes into account all paragraphs of the alleged offences (routine, non-routine, factual, and (redundant) delict description paragraphs) and all paragraphs of the opinion of the court (routine, non-routine, factual, and principle paragraphs). Routine paragraphs were eliminated in the first step of the abstracting process (see chapter 7). In the original case text, the expert marked the paragraphs to be included in the abstract. The above metrics are used to compare this manual abstract with the final output of the system. The evaluation gives insight into the combined use of deep and shallow techniques. It also evaluates how well the

system performs in extracting principle paragraphs considering the noise of routine paragraphs and factual considerations.

In case of the alleged offences, the errors of the initial structuring influence the results. The results are still very good with a recall and precision of 82%. The low fallout rate (9%) indicates that the system chooses correct responses even with a high number of possible responses. In case of the opinion of the court the errors of the initial structuring phase also influence the results. The system finds an important part of the legally relevant principle paragraphs that were intellectually attributed (75%), but generates too many paragraphs (overgeneration of 55%). Such a large overgeneration necessarily decreases precision (33%). Precision is computed as the proportion of correct answers in all the answers generated. The overgeneration concerns some routine grounds and many factual considerations.

SUMMARY OF CRIMINAL CASE
NAME OF CASE = /users/sien/testset/algemeen/gg/g2
DATE = November 10, 1992.
COURT = CORRECTIONAL COURT LEUVEN
REPRESENTATIVE PARAGRAPHS OF THE OFFENCES=

REPRESENTATIVE PARAGRAPHS OF THE OPINION OF THE COURT=
Whereas ... the accused admitted in his statement of 18.11.91 that he did not draw up any labor regulations, nor written labor contracts for part-time workers;

REPRESENTATIVE KEY TERMS OF THE OPINION OF THE COURT=
labor contracts part-time

REPRESENTATIVE GROUNDS =
ON THESE GROUNDS and implementing the articles 1384 of the Civil Code; 38-40-65 of the Criminal Code; 1–4-25.1-27-28 of the law of April 8, 1965, that establish the labor regulations; 2 of the Collective Labor Agreements of 27.2.1981 concluded in the National Labor Council, regarding some regulations of the Labor Law towards part-time work, declared as generally binding by the Royal Decree of 21.9.1981; 11 bis 2 of the law of 3.7.78, regarding the labor contracts; 56.1-57-59-60 of the law of 5.12.68, regarding Collective Labor Agreements and Joint Committees

Figure 3. Example of a case summary (translated from Dutch).[2]

SUMMARY OF CRIMINAL CASE
NAME OF CASE = /users/sien/testset/verli
DATE = September 16, 1992.
COURT = CORRECTIONAL COURT LEUVEN
REPRESENTATIVE PARAGRAPHS OF THE OFFENCES=
by use of violence or threat, to have destroyed or damaged others movable property, namely doors, bottles, glasses, chairs, tables, crates of beer and coke belonging to ... and ... .
under the circumstances that the facts were committed in association or in gang, and that ... was the leader or the fomenter of the gang.
To have committed assault and battery to ..., causing illness or inaptitude for the accomplishment of personal work;
To have committed assault and battery to ...
By way of gestures or symbols to have threatened ... with offences against his person or property, punishable with an imprisonment imposed by a Crown Court.
REPRESENTATIVE PARAGRAPHS OF THE OPINION OF THE COURT=
Whereas … claims, without foundation, that he may be responsible for what happened during the so-called first brawl, close to the disco, but that he was not involved with the incident that occurred a little later close to the bar; whereas this clearly was a continued group incident with the first accused acting as the most violent person, to the extent that he should be considered at that moment as the main fomenter,
REPRESENTATIVE KEY TERMS OF THE OPINION OF THE COURT=
group incident brawl
REPRESENTATIVE GROUNDS =
ON THESE GROUNDS and implementing the articles 1382 of the Civil Code, 38-40-44-50-65-66-528-529-79-80-84-327-329-392-398/1-399/1 of the Criminal Code;

Figure 4. Example of a case summary (translated from Dutch).[3]

The result of applying the two abstracting steps to the criminal case is a case profile ("index card") that on average is 20% of the original case text (Figures 3 and 4). The size of a case text varies from one to about 12 pages. For long texts, the reduction in text size is larger. The summaries of the alleged offences are exempt of redundancies and are reduced to an average of 78% of the size of their original texts. The texts of the opinion of the court are condensed to their representative paragraphs and are reduced to an average of 49% of the original texts. The reductions in size are close to the text reductions proposed by the expert.

Table 2. Average results of a "legal" evaluation of the abstracting of alleged offences and opinion of the court.

	Effectiveness measures			
	Recall	Precision	Overgeneration	Fallout
Alleged offences	0.817327	0.817422	0.117243	0.091463
Opinion of the court	0.746371	0.330498	0.545124	0.214610

By doing this, the SALOMON system could *simulate part of the practice of human abstracting*. When looking at the cognitive process of abstracting (see chapter 3), it seems that some aspects could be automated. They include the identification of the case type, the structure of the information, deletion of redundant and insignificant information, and selection of thematically relevant text units and key terms. Our research demonstrates that this part of *intellectual abstracting* can effectively be *simulated.* However, part of the intellectual process, which involves *interpretation*, was *out of reach*. The shallow techniques employed for theme recognition do not allow the discrimination between principle and factual grounds in the opinion of the court. Principle grounds are the paragraphs of the opinion in which the judge gives general, abstract information about the application and the interpretation of some statutes. The identification of principle grounds requires interpretation based upon contextual information, to be found within, as well as beyond, the text of the case: other statutory provisions, legal principles, and multiple social customs and norms (Uyttendaele et al., 1996, 1998). The system recognizes 75% of the principle grounds, but generates also representative paragraphs of the factual grounds. So, it is not possible to automatically discriminate leading cases with the statistical techniques.

Nevertheless, a system like SALOMON can simplify the lawyer's job a great deal (see Uyttendaele et al., 1996, 1998). It does not provide the user with ready-made answers to complicated legal cases. But, it directs the lawyer towards documents where the answer must be found (cf. Zeleznikow & Hunter, 1994, p. 73). SALOMON is a tool telling the lawyer what the law is in a certain case, what crimes are committed, and about which topics the judge motivates.

5. CONTRIBUTIONS OF THE RESEARCH

The research presented in this chapter certainly makes a contribution to automatic abstracting of a text's content.

The topic structure of elaborated offences and opinions of the court is automatically recognized while building upon shallow techniques, recently developed in the domain of information retrieval. However, the use of *cluster algorithms based on the selection of representative objects* is new in this context.

Simple statistical methods for text summarization extract sentences that are considered as important because they contain high-weighted terms. More advanced methods aim at automatically *determining the topic structure of a text based on patterns of lexical connectivity* in the text. They group sentences, paragraphs, segments with a fixed number of sentences, or segments with a fixed number of adjacent words based upon common term usage. Text units are grouped when their mutual similarity between their term vectors exceeds a certain threshold. We refer to chapter 6 for a detailed overview of these methods. According to Prikhod'ko and Skorokhod'ko (1982) sentences related to a large number of other sentences by means of their content terms are highly informative and are prime candidates for extraction and inclusion in a summary. After we finished the SALOMON project in 1996, Salton, Singhal, Mitra, and Buckley (1997) proposed algorithms to extracts relevant sentences from a group of related sentences with rather poor results (maximum 46% overlap with manually prepared extracts of the texts). We propose alternative algorithms for grouping the sentences or paragraphs according to topics and for identifying representative sentences or paragraphs in a group of related sentences or paragraphs, which yield very satisfying results.

An important technique for grouping related objects is *cluster analysis*. In text-based systems, clustering algorithms have been employed for grouping similar documents on the basis of terms that co-occur in the documents, providing an efficient search structure for large document collections. Clustering techniques are also useful for detecting similar documents to the ones relevant for a query. Hierarchical methods produce a hierarchic structure of the data (for their use in text-based systems see Voorhees, 1986, and Willett, 1988). Non-hierarchical methods result in a partitioning of the data set by clustering the data around cluster representatives. The non-hierarchical methods used in text-based systems (single-pass method and methods based on the construction of central points) employ centroids as representative objects of the cluster (Salton, 1971, p. 223 ff.; cf. Anderberg, 1973, p. 156 ff.). The *centroid* is calculated as an

average representation (e.g. average vector of the objects), which is dynamically computed as the cluster structure is built. A drawback of these algorithms is that the results depend on the order of the objects in the input. The centroid changes each time that an object is added to the cluster. Moreover, these methods do not deal with the search for an optimal number of clusters, nor do they identify a representative cluster object as the most centrally located object of the cluster.

The use of *cluster algorithms based on the selection of representative objects* is new in text-based applications. They have a definite potential in recognizing the topics of a text and identifying relevant sentences or paragraphs to be included in the abstract. These algorithms have many advantages. The *k*-medoid method produces a natural clustering. This was important in order to obtain a balanced summary of the opinion of the court that contains a representative paragraph and key terms regarding each topic treated. The algorithms also provide the possibility to identify informative text units relevant for abstracting purposes.

Some initial knowledge about the thematic structure of the alleged offences and opinion of the court allowed one to choose the effective clustering algorithms. Findings about the thematic structure of different text genres and discourses may induce additional research into the statistical recognition of the topics of a text.

The combined uses of shallow statistical techniques discussed in this chapter and the deeper techniques that rely upon a text grammar discussed in the previous chapter have proved their effectiveness. According to Sparck Jones (1993) progress in automatic abstracting might be realized along two directions. First, *text structure* of the discourse type is important when accessing the content of a text. Second, the progress made in information retrieval, especially the current refinement and sophistication of *statistical indexing techniques*, might be fertile for abstracting texts of unrestricted domains. It is along these two directions that we have developed the SALOMON project and effectively have made progress in automatic abstracting.

Automatic abstracting of legal texts is barely researched. There is the FLEXICON system, which automatically generates a summary of a legal case to be used in a database for case retrieval (Gelbart & Smith, 1995). For identification of significant text units, it relies upon location heuristics, frequency occurrences of terms, and the use of indicator phrases. The system was not systematically evaluated. The more advanced techniques that we propose certainly make a contribution to the automatic processing of legal texts and to the research field of artificial intelligence and law (cf. Moens et al., 1997).

6. CONCLUSIONS

A growing amount of electronically available legal cases enlarges the need for effective access to these documents. The automatic generation of case abstracts is one way to ensure their accessibility. SALOMON extracts relevant text units from the case text to form a case summary. Such a case profile facilitates the rapid determination of the relevance of the case, or may be employed in text search.

After an initial selection of relevant text passages from the case texts, the passages are further condensed with the help of clustering techniques. Cluster algorithms based on the selection of representative objects provide the possibility to identify informative text units that through their lexical patterns are linked to other text units. As a result, redundant information is deleted from the delict descriptions and thematically coherent text pieces of the argumentation of the judge are identified. The techniques proposed contribute in automatic text structure recognition, theme identification, and abstracting. The algorithms have the following advantages. They do not rely upon the order of input, and identify a representative text unit in a cluster of related units that is informative about the content of the cluster. The *k*-medoid method produces a natural clustering. This is important in order to obtain a balanced summary that contains representative paragraphs and key terms regarding each topic treated.

---

[1] Tree full stops indicate a personal name erased for privacy reasons.

[2] The term “labor contracts” is a compound noun in Dutch and is written as a single word (“arbeidsovereenkomsten”). The system did not find the offences in this case.

[3] The term “group incident” is a compound noun in Dutch and is written as a single word (“groepsgebeuren”).

Chapter 9

THE CREATION OF HIGHLIGHT ABSTRACTS OF MAGAZINE ARTICLES

1. INTRODUCTION

An important Belgian publisher offers *magazine articles on-line* against payment, The magazines cover news stories on a variety of subjects. In a traditional commerce context, the potential buyer of a magazine can look at its cover pages, and even browse the publication for a short time before deciding to purchase it. In an electronic environment, the selection is more complicated.

The work reported here is part of the *Media On Line project*, which concerns providing access to the products of a Belgian publisher. An important part of the project regards automatic text analysis of magazine articles in order to represent them adequately for on-line selection. A first selection technique involves browsing a database of article abstracts before deciding which articles to buy. A "*highlight abstract*" consists of clippings extracted from the article text and aims at attracting the reader's attention. It must deal with the main topics of the text and instantly rouse curiosity and interest in the magazine article. Creation of abstracts by professional abstractors is expensive and slow, which brings about the need for their automatic creation. A second selection technique regards the categorization of the articles with subject descriptors. The descriptors are used to effectively route the articles to subscribers of the magazine who are interested in articles on specific topics. The research regarding this second selection technique is presented in the next chapter.

This chapter deals with the design and implementation of a system for creation of highlight abstracts (see also Moens & Dumortier, 1998). The research focuses upon the efficacy of using knowledge of text structures and their signaling linguistic cues as implemented in a text grammar. The proposed formalism, which is used for representing the grammar, is discussed in chapter 7. In this chapter we demonstrate that it captures different types of knowledge for text abstracting and is portable and flexible to adapt to changing text types and abstract needs.

This chapter is organized as follows. We describe the text corpus and the desired output of the system. A short discussion of the methods is followed by an evaluation of the results and by a description of the contributions of the research.

2. TEXT CORPUS AND OUTPUT OF THE SYSTEM

The articles come from the magazine "Knack" and cover broad subject domains (e.g., politics, economy, fashion, sports, and arts). They are written in Dutch. They are of varying length, ranging from a few paragraphs to multiple pages, All articles belong to the general discourse type of written news stories and cover both hard news and feature articles (cf. Bell, 1991, p. 14). *Hard news* reports accidents, crimes, announcements, discoveries, and other events that have occurred or come to light since the previous issue of the magazine. Articles of the column "België" belong to this category. *Features* are longer articles, and do not necessarily cover immediate events. They contain elaborate stories on a variety of topics. The stories provide much background information of the topic and sometimes carry the writer's personal opinion. The majority of them have a narrative character, but they do not only recall past events, but also interpret and explain events. The articles of the column "Document" are representative feature articles. Some articles regardless of the column are told in the form of an interview.

The purpose of the abstract is to "highlight" certain aspects of the article in a way that its user is maximally interested in buying the article. Browsing the article abstracts especially aims at the casual user of the information. Reading the abstract in the electronic environment must be part of simulating the leafing through of the magazine article at a bookshop. We are here not concerned with the final layout of the abstract, which is very important for this purpose, nor in the addition of images and sound in its final version, but focus upon the textual content of the abstract as a mean of rousing the curiosity in the complete article.

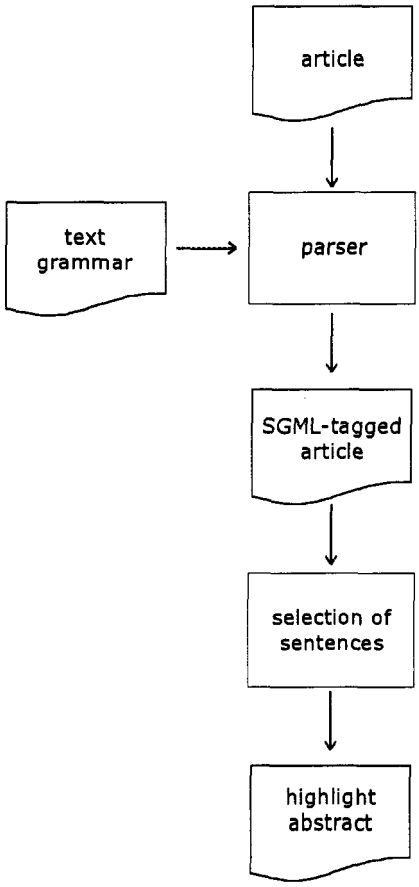


Figure 1. Architecture of the demonstrator for creating highlight abstracts.

The highlight abstract is *indicative of the content* while *stressing certain information*. The abstract must suggest the main topics of the article without going into too much detail, which might make the reading of the complete text superfluous. Besides being indicative it must be appealing for a potential buyer of the article. The abstract consists of clippings of text, i.e., sentences and statements extracted from the text. It preferably contains short, easily readable sentences, which, for a correct interpretation, do not rely upon the context of the surrounding article text.

A demonstrator is built for assessing the value of the methods employed (Figure 1). Except for the text grammar module, the demonstrator is ported from the tool that was successfully employed in the SALOMON project for an initial categorization and structuring of criminal cases (see chapter 7; Moens & Uyttendaele, 1997). The system is built in the programming

language C on a Sun™ SPARC station 5 under Solaris® 2.5.1. Its major components are a knowledge base containing the text grammar(s) of the text type(s) and a parser for analyzing the texts based upon the text grammar. We reused this abstracting tool but with a different text grammar, which reflects the discourse properties of news stories. The result of the parsing based upon the text grammar is an article tagged in SGML(*Standard Generalized Markup Language*)-syntax. The identified text segments axe marked with the appropriate category tags. From the tagged article text, statements and sentences to be included in the highlight abstract are easily selected.

3. METHODS: THE USE OF A TEXT GRAMMAR

As seen in chapter 3, automatic abstracting consists of three steps. A text analysis step aims at identifying the content of the text. It is followed by the selection and possible generalization of the textual data that are relevant to be included in the abstract. A final step concerns text generation in which the text of the abstract is edited or rewritten. Because the abstracts of the magazine articles contain sentences and statements extracted verbatim from the texts, the methods here regard the *text analysis* and *information selection step* of the summarization process.

3.1 Linguistic Background

Discourse structures is important when analyzing text for summarization (Endres-Niggemeyer, 1989; Sparck Jones, 1993; Moens, Uyttendaele, & Dumortier, 1999b). In chapter 2, we extensively elaborated on discourse analysis including a micro and macro level description of texts.

News discourse has extensively been studied by van Dijk (1985, 1988a, 1988b), Bell (1991), and Fairclough (1995). On a macro level, news discourse exhibits several structures, which are created by the writer of the text in order to attain his or her communicative goal including the emphasis on certain aspects of the content and to cue the reader into the writer's perspective (cf. Coulthard, 1994).

News stories in the written press have a conventional *schematic structure* or *superstructure* (van Dijk, 1988b, p. 49 ff.; Bell, 1991, p. 169 ff.), which consists of ordered (at least in part) components or segments. The text of a news story is typically composed of a headline, a lead, an attribution, and the body of the story. Although it may pick up on a minor point of the story, the *headline* usually abstracts the main event of the story. So, *headlines* are not just a summary but part of the news rhetoric whose function is to attract the

reader. The *lead* paragraph establishes the main points of the story. Heading and lead form the journalist's abstract of the story. Also, the lead focuses the story in a particular direction. The *attribution*, which is situated after the lead before the main part of the story or is included in the lead, contains the general setting of the story (source, actors, time, and place) as well as the source of the information. A story is composed of one of more episodes, which in turn consist of one or more events. An *episode* or event can have its own detailed attribution and setting. The *events* together with their follow-up, commentary and background form the main part of the news story. *Follow-up* covers any action subsequent to the main action of an event. *Commentary* provides the journalist's or news actors' observations on the action. *Background* covers context and previous events. The segments of the superstructure may be signaled by specific linguistic cues such as the use of cue phrases (e.g., a comment section may start with "in my opinion") (cf. Liddy, McVearry, Paik, Yu, & McKenna, 1993).

The *thematic structure* of the news story concerns its overall organization in terms of the topics and subtopics. In news discourse the schematic structure typically parallels the thematic structure. The more general topics come first in the story to be followed by more detailed information (van Dijk, 1985; Fairclough, 1995, p. 30). The main topics appear in the lead and the attribution. Topics become more specialized as the text progresses. The topics of a text are closely related to the surface linguistic phenomena of the text: They can be cued by lexical items or by the position in the text. In news stories, topic sentences are often located as the first sentence after a subheading or as the first sentence of a paragraph.

News stories also contain typical rhetorical features that signal *rhetorical structure.* These features are used to attain a certain communicative goal. In the stories there is always the tension between information and entertainment, between the semi-technical and popular character. Journalists use common rhetorical devices to attain this effect (Bell, 1991, p. 204 ff.; Fairclough, 1995, p. 32 ff.). The stories employ a so-called conversational language, which makes news interesting and more accessible to people, The conversational character is expressed in features such as interrogative and imperative clauses and by the direct representation of the talk of others. Quotations ("quotes") in news stories are supposed to be brief, pithy, and colorful and to add a flavor of the eyewitness and direct involvement. Moreover, journalists are inclined to use alliteration, punning, and metaphorical language.

After intellectually analyzing about 100 magazine articles of the Belgian publisher, the above discourse structures and signaling linguistic cues were confirmed.

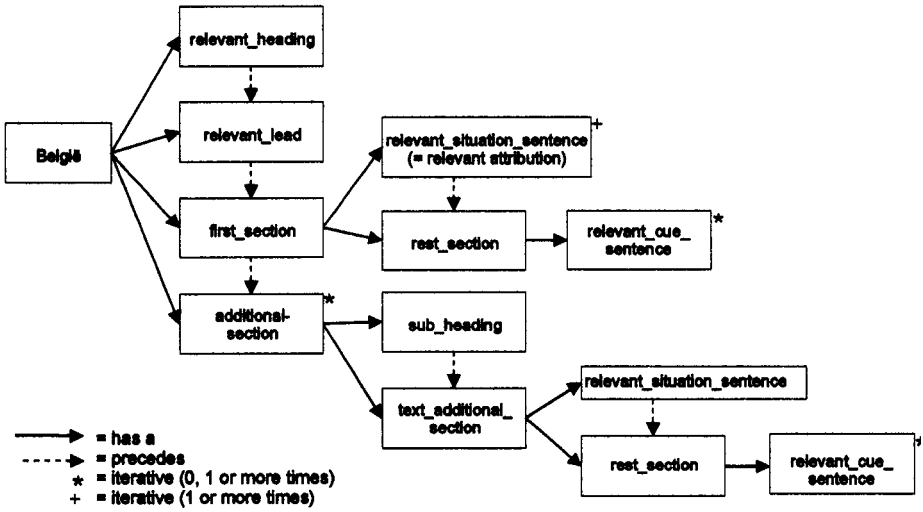


Figure 2. Example of a representation of a hard news article.

3.2 Knowledge Representation

To incorporate knowledge of the discourse patterns of news stories into text processing applications, we need an adequate knowledge representation. The *knowledge base* that we propose employs a *domain-independent formalism.* It allows representing text structure in the form of a text specific grammar. The use of a *text grammar* is appealing for several reasons (Moens et al., 1999b), the most notable being that many text types can be decomposed into a limited set of constituents that combine with one another in regular ways.

The formalism for representing the text grammar has been described in chapter 7 p. 161 ff. (see also Moens & Uyttendaele, 1997). The grammar includes the major semantic components or segments of a text (e.g., paragraphs, sentences), their attributes (e.g., optionality), and the relations between them (e.g., hierarchical, sequential) or with typical lexical patterns (e.g., being classified by, being delimited by). Segments of a text are recognized by their typical order or by identifying occurrences of various strings referred to as word patterns.

A set of articles was manually analyzed. *Acquisition and implementation of the knowledge* in the text grammar formalism required one man week. The knowledge *for creating highlight abstracts* relates to two slightly different grammars (one for the hard news articles of the column “België” and one for the feature articles of the column “Document”) with about 25 definitions of segments categories, 12 word patterns organized in 8 classes, about 40 relations between text segments, and 13 relations between segments and pattern classes, After an initial testing, the grammar of the column "Document" was slightly adapted.

The structural representation of the articles breaks them up into segments, some of which are useful to include in the highlight abstracts. It roughly represents the schematic structure of the magazine articles (Figure 2). Headline and lead are important to include in the abstract, because of their property of catching the attention for the story and establishing its main points. The parallels between the schematic and thematic structure in the stories are taken advantage of. The topic sentences of the attribution and at the beginning of subsections of the articles (in Figure 2 called “relevant_situation_sentence”) are important to include in the highlight abstract, because they treat the main topics of the article. Lexical cues that indicate hot topics are also crucial. For instance, the semantic class of lexical items that signal “corruption” is important for this purpose, when summarizing current articles about Belgian politics. Rhetorical cues are also employed to define cue sentences.

It is possible to selectively represent knowledge about discourse patterns that are relevant to the task at hand and to determine the level of analysis desired. Consequently, multiple views of a text are possible, each conforming to different needs. The simple text grammar can be refined or altered in order to cope with changing content requirements of the abstract.

THE CHOICE OF DEPREZ

Gérard Deprez has resigned as PSC-chairman, but not without arranging his succession.

It was already some time in the air. The ones that regularly kept up with him, brought the news that PSC-chairman Gérard Deprez was fed up with it. “Of this government, of the socialists, of some of his party members” – especially of the last ones. . . .
The PSC-chairman had to give precedence to this party member *Melchior Wathelet*. . . .
Joëlle Milquet considers her candidacy. . . .

Figure 3. Example of a highlight abstract of a hard news article of about 1 and a half pages (translated from Dutch). A page contains about 800 single words.

THE BLOCKS stand NEXT TO THE CRADLE

Two thirds of the consultation offices of Child and Family come in catholic hands, one quarter in socialistic hands. Sectarianism at highest, finds Patrick Vankrunkelsven (VU).

"Sectarianism is expensive and hypothecates a good dispersal of the consultation offices," says *Patrick Vankrunkelsven,* general practitioner and vice-chairman of the VU. „I don't understand the critique, because sectarianism doesn't scare the population. Moreover, it is not the aim of Child and Family to break up sectarianism, " says *Lieven Vandenberghe,* general administrator of the public institution that takes care of the health of children to three years old.
With fewer offices, the public institution wants to reach more deprived families .. .
Vankrunkelsven went with the volunteers of Laakdal to chairman *Paula D'Hondt* (CVP) of Child and Family ...
Vankrunkelsven fundamentally regrets that Child and Family perpetuates sectarianism and that general practitioners are not involved in the project ... Do we promote sectarianism with this? ... That there are two consultation offices next to each other in Geel? ... But, if this is so, is the landscape not more multi-colored? . . .

Figure 4. Example of a highlight abstract of a hard news article of about 2 pages (translated from Dutch).

3.3 Parsing of the Text and Generation of the Abstract

The deterministic *parser* is composed of two major modules (see chapter 7, p. 164 ff,): a push down automaton for analyzing the nested structures of text segments defined by a context-free grammar and a finite state automaton that recognizes regular expressions in the text. The activation of a segment frame may depend upon a positive evaluation of a function attached to the frame. The function can evaluate a simple production rule or be a complex procedure. In this way a restricted and controlled form of context dependency can be implemented. The parser focuses on finding the text components defined in the text grammar, while neglecting the remainder of the text and can thus be regarded as a partial parser.

After a segment is found, its begin and end positions in the article are marked with the segment name. Tags in *SGML*-syntax are assigned. Except for the insertion of category tags, the parsing does not structurally, lexically, morphologically, or syntactically alter the original article.

WHITE SMOKE IN THE REYERSLANE

Last week, the Flemish government presented Bert De Graeve being the new BRTN-manager. Does this give the house confidence?

Last week on Tuesday night, *Bart De Schutter*, professor at the Vrije Universiteit Brussels and chairman of the BRTN, received a call on his mobile phone. On the other side of the line, a numb reporter of the newspaper *De Morgen*, who spread her bed before a Brussels restaurant. There, according to the newspaper's informer, the new BRTN-manitoe would be represented. Where is De Schutter, the furious reporter wanted to know? ...

His evening was completely successful, when he could tell the anecdote how in an assault of shrewdness he had led the press up the garden path. …

De Morgen, thus missed a *scoop*, but more interestingly than the name itself, was the fact that, for one time, the person concerned wasn't shining in the media one second earlier than foreseen...

With Bert De Graeve, BRTN gets a so-called a-political manager, although the director-general of the television *Jan Ceuleers* rightly noticed that "anyone has a political conviction, especially someone who has a degree." ...

In the meantime, the new delegate manager of the BRTN considerably scored at his first public appearances ...

In the weekend, the "old guard", brutally left behind, made itself heard by mouth of director-TV Ceuleers. ... Who else than Jan Ceuleers can better recognize this approach? ...

Bert De Graeve's task seems falsely simple, when conceived in one cryptic and hollow slogan: " to give the broadcasting more strength." ...

Is it, for instance, opportune to put the so-called general and technical services on the market as a completely autonomous unit? …

Figure 5. Example of a highlight abstract of a hard news article of about 3 pages (translated from Dutch).

The result of the parsing of the article is a text in which the segments defined in the text grammar are identified. Some of these segments are relevant to include in the abstracts and have received a special naming in the text grammar (e.g. in our application: these segments have the prefix "relevant" in their name: see Figure 2). From the article with SGML mark-ups, those segments that are relevant to include in the highlight abstract are easily selected. The recognition of the other segments was needed as a necessary intermediate step in recognizing the texts units to be included in the abstract. The sentences and statements extracted are not altered in any way. Even, the original layout markups are preserved. When the sentences do not follow each other in the original texts, they are separated by three full

stops in the text of the abstract. The abstracts are about one tenth to one twentieth of the length of their original texts.

The research concentrated upon text analysis and extraction of statements and sentences. No effort is made to generate a better abstract by rewriting these sentences and statements. A complete reformulation endangers the original flavor and style of the language of the journalist who wrote the article. However, some reformulation seems useful when refining the current system, Highlight abstracts benefit from short, to-the-point sentences. We might identify foreground information from background information in some of the longer sentences. Then, background information, such as descriptive relative clauses, could be deleted in the extracted sentences.

THE ART COMES FIRST

During his stay in Berlin, Paul van Ostayen shouldered a large task: on his own he wanted to salvage the German expressionism from a sure destruction. A publication.

The business of the German plastic artist *Fritz Stuckenberg* did not prosper in 1919. In the first half of that year, because of a financial dispute he broke with his steady representative in Berlin, *Herman Walden*, who exploited by far the most important expressionistic art gallery in the German capital. There were often complains about Walden's practices. ... Comment of *Paul van Ostayen*, exile in Berlin and good friend of Stuckenberg, about Walden: "villain and exploiter of artists." ... And Stuckenberg had reasons to think that Walden had "a holy fear" of Van Ostayen, who in his terms knew Walden as "a dangerous rival": he could get fair prices for Stuckenberg's work. ...

"One artist after the other leaves Der Sturm, " ascertains Van Ostayen. ... There was more than purely an uproar of aversion for Walden's practices – were they ever been different? Corruption and self admiration are only one side of the picture. ... The visit had convinced van Ostayen: "The Bauhaus signifies absolutely nothing." ... Social security for the painters and their children: C*'est le commencement de la fin.* And a production, dear God ! From Breslau to Berlin: one museum for expressionism, cubism, and futurism ! !

Van Ostayen readily had a solution: again starting from scratch, "back up the path of simplicity to the most severe aspiration." ...

For Muche there was a problem, he certainly agrees with the idea that merchants "are not allowed having anything to do with art", but he had still a contract with Herwarth Walden, which finished at the end of October 1920, to fulfill. ... Nothing to worry: van Ostayen's brother *Constant !* ...

Figure 6. Example of a highlight abstract of a feature article of about 4 pages (translated from Dutch).

4. RESULTS AND DISCUSSION

A sample of 40 magazine articles of the above text corpus was chosen at random This test set is distinguished from the article set employed for the construction of the text grammar. It is composed of 30 hard news articles, 3 feature articles, and 7 interview articles. Only two of the interview articles feature a hard news topic. The resulting abstracts show that the proposed techniques are promising.

The 40 summaries (e.g. Figure 3-6) indicate that plausible highlight abstracts can be generated by analysis of the original article texts based upon a simple text grammar with discourse patterns that are typical for news stories. Especially, the abstracts of *hard news articles* are satisfying. We could rely upon the multiple discourse studies of news stories. The *feature articles* are more difficult to summarize. The events do not necessarily follow a temporal sequence. They are rich in flashbacks, commentary, and background, and may expose unexpected twists. They sometimes interweave different stories. As a result, the clippings extracted may be wrenched from different contexts, which may confuse the reader of the abstract (cf. Figure 6). In the experiments we found that often only introductory information from the beginning of the article could be reliably extracted. It is clear that we need more studies about the discourse patterns and communication structures of this kind of article. Creating highlight abstracts from *interview articles* resulted in plausible abstracts. Some refinements are possible. Besides some introductory information, questions that are asked in the interview are interesting to rouse curiosity of a potential buyer of the article. But a selection of questions is necessary, in order not to go into much informational detail and not to compromise the compactness of the highlight abstract. For selecting the right questions, we need more discourse studies about the communicative value of different types of questions and their signaling cues. Also, statistical techniques for topic recognition based upon word distributions are useful to discriminate questions that treat different topics.

Evaluation of automatic summarization is a difficult task. An *intrinsic evaluation* (Sparck Jones & Galliers, 1996, p. 19 ff.) traditionally measures the similarity between automatically generated summaries and human prepared ones in terms of quantitatively measuring the completeness, correctness, and superfluity of the information in the summaries that are automatically generated. However, a correct highlight summary is difficult to establish. There may be more than one good solution. The Belgian publisher compared the 40 abstracts with abstracts of the same texts that were made with a commercial abstracting tool. The commercial tool extracts

sentences from the texts that contain highly weighted terms. Our solution was preferred over the commercial tool and a more refined version of the text grammar is now integrated into the document management tool of the publisher.

In case of highlight abstracts, it is important to evaluate whether the summary meets the user's need and to assess the legibility of an abstract independently from the source text (cf. Sparck Jones & Galliers, 1996, p. 19 ff.). Such an *extrinsic evaluation* judges the quality of the summary, based on how it effects the completion of the task it is intended for. The creation of highlight abstracts may be judged successful if an increase in sales of the electronic articles can be obtained, when the selection of articles is based on these summaries.

Evaluation also concerns assessment of the qualities of the formalism in representing the knowledge for abstracting.

According to Sparck Jones (1993), knowledge representations for abstracting may capture linguistic knowledge, domain world knowledge, and communicative knowledge (Figure 7). All three types of information give a very different characterization of the text. The *linguistic knowledge* that deals with the schematic, rhetorical, and thematic structures and their signaling phenomena such as ordering and lexical cues, is especially useful for abstracting purposes. This knowledge embodies standard, known ways of organizing texts that are conventionally associated with seeking to achieve certain communicative effects. *Domain world knowledge* deals with the representation of the concepts typical for the subject domain treated in the text. *Communicative howledge* is part of the contextual knowledge of the communication process. It deals with the representation of the intentional structure recognized by the reader of a text or in case of abstracting with the representation of the focus of attention of the abstract (cf. “attentional state” in Grosz & Sidner, 1986). In the proposed formalism, the three kinds of knowledge can be represented in an integrated way. The formalism especially aims at representing text structures as an ordered set of composing segments (network of segment frames) and as flagged by lexical and other surface cues (pattern frames). Domain knowledge can be represented by the word pattern frames that represent a semantic class and its variant patterns. For creating the highlight abstracts, domain knowledge was only minimally present. The text grammar also specifies the communicative knowledge. An abstract can have a special focus of attention, which according to the task of the abstract may only be part of the creator's communicative goal. The formalism allows the defining of different views of the text according to the focus of attention of the abstract.

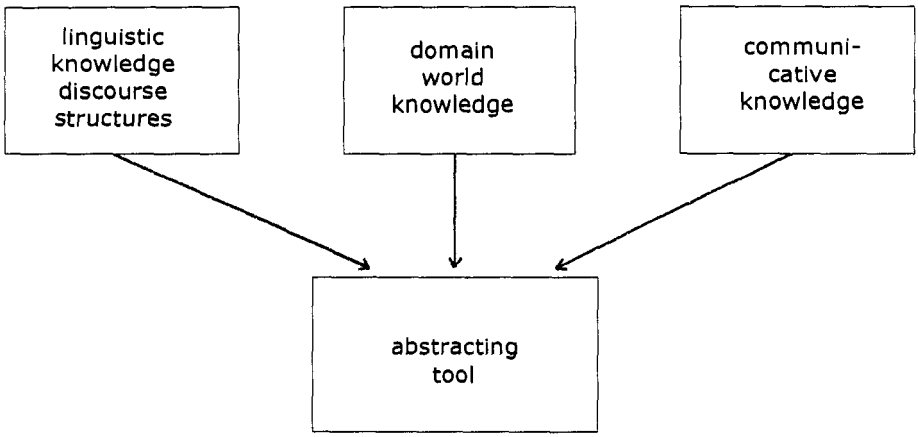


Figure 7. Important knowledge sources in automatic abstracting (cf. Sparck Jones, 1993).

In order to be useful, the formalism must be *portable* and *flexible*, so that it can be used for representing different text types and abstracting needs. The system was previously successfully applied for abstracting Belgian legal cases and was used to structure the text into relevant and irrelevant text passages and to extract specific data from the case text (e.g., relevant statute citations) (see chapter 7 and Moens & Uyttendaele, 1997). The different types of text segments allow the analysis of the text on a passage (text block), paragraph, sentence, clause, or even word pattern level. This makes the formalism flexible to comply with different text types and information structuring needs.

Parsing speed is acceptable. Analysis of an article and output of its abstract take 3 to 8 seconds on the Sun™ SPARC station 5 (85 Mhz).

The text grammar approach has been proven useful for information extraction from texts and for abstracting texts, but these are not its sole *applications* (Moens et al., 1999b). It has many applications in text indexing. For instance, subparts of documents can be identified which are likely spots to contain relevant content. From them, index terms, which are used as search terms in information retrieval or as features for learning text classifiers (see next chapter), can be extracted or highly weighted. Other significant applications of text grammars are the text drafting and generation

systems that aim at enhancing the communicative value of texts and at making texts more easily understandable by machine.

5. CONTRIBUTIONS OF THE RESEARCH

Automatic summarization of texts is a difficult task, especially when the texts cover heterogeneous subjects, which is the case with magazine articles. As it is explained in chapter 6, many successful summarization systems heavily rely upon *domain knowledge* of a restricted subject domain. The use of *discourse structure* in automatic summarization has always been minimally present in summarization systems, but it is becoming of increasing importance (cf. Hobbs, 1993).

First, our research proves the importance of discourse patterns in automatic abstracting. These include the *schematic structure or superstructure* of a text and its signaling linguistic cues, the *rhetorical relations* hinted at by cue words and indicator phrases, and the thematic structure of a text. Liddy et al. (1993) use an explicit model of the schema of the news story text type for automatically structuring Wall Street Journal articles into discourse segments. Recognition of the segments heavily relies upon their order of precedence in the article texts and their signaling lexical cues. We were able to combine this knowledge of news schemata with knowledge of the thematic structure and rhetorical cues in order to generate plausible abstracts of magazine articles that reflect their main topics while rousing the curiosity for the full articles. In addition, our research demonstrates that acceptable abstracts can be automatically generated from texts discussing a large variety of subjects by exploiting the knowledge of the discourse patterns.

Second, our research proposes a domain-independent formalism that allows representation of texts of different types. As already demonstrated when abstracting legal cases (chapter 7), the proposed text grammar formalism integrates linguistic knowledge of the discourse structures, their ordering and cues with domain and communicative knowledge. Using the formalism for abstracting completely different texts, namely magazine articles, only confirms the usefulness of the approach. There is also a growing interest in tailoring abstracts to the specific needs of users (Sparck Jones & Endres-Niggemeyer, 1995). The text grammar allows representation of different views of a text and focusing on specific information.

6. CONCLUSIONS

This chapter demonstrates that the typical discourse patterns of magazine articles can be implemented in a text grammar and employed to automatically create highlight abstracts of the articles. The writer of a text employs specific discourse patterns (schematic, rhetorical, and thematic) so that a reader maximally discovers the text's message. The patterns also guide the reader in interpreting the passages of the text. A reader can likewise approach a text with various structural expectations. It is this shared knowledge that is highly valued in automatic summarization of texts and in identifying certain information in it.

The text grammar formalism proposed comprises an initiative to integrate different discourse structures and their signaling cues together with valuable domain and communicative knowledge. Moreover, the formalism is portable to different text types and flexible enough to accommodate different abstracting needs.

This page intentionally left blank.

Chapter 10

THE ASSIGNMENT OF SUBJECT DESCRIPTORS TO MAGAZINE ARTICLES

1. INTRODUCTION

The *Media On Line project* regards automatic text analysis of magazine articles in order to represent them adequately for on-line selection. A first selection technique involves browsing a database of highlight abstracts before deciding which articles to buy. Creation of the highlight abstracts was the subject of the previous chapter. A second selection technique regards the categorization of the articles with subject descriptors. The descriptors are used to effectively route articles to magazine subscribers who are interested in electronic articles on specific topics. A fast routing of articles immediately after their publication is important, hence the interest in automating the process. Categorization of the articles is subject of this chapter.

The research here presents experiments with different text categorization algorithms that were tested upon the texts of magazine articles written in Dutch. The behavior of different text classifiers and the results are explained given the properties of the texts in the subject domains. The algorithms learn the classification patterns from example texts that are manually classified. The χ^2 test is applied in a novel way when constructing a category weight vector with satisfying results. There is a strong focus upon effective selection of content terms and proper name phrases. We investigate whether the techniques of selecting negative examples and of a divisive clustering of the positive examples of the text class can improve the results. An important

constraint of the research is the limited number of positive examples in a text class available.

This chapter is organized as follows. We describe the text corpus and the desired output of the system. The methods are discussed in detail and are followed by an evaluation and discussion of the results. We finish with the contributions of the research.

2. TEXT CORPUS AND OUTPUT OF THE SYSTEM

The more than 2650 *articles* of the text corpus were published in 1998 in magazines such as "Knack", "Weekend Knack", "Trends", and "Cash!". They are written in Dutch and are very heterogeneous in content and structure. The articles cover a variety of subjects in domains, such as politics, economy, finance, life style, arts, sports, and many others, and often interweave different subject domains. The articles belong to different columns of the magazines. This is reflected in their structure. Most of them follow the schema of the written news story, but other schemata are present, such as a list of film titles with explanatory sentences, or a simple address of a restaurant. A few articles are so-called satellite articles: They are small texts that elaborate on a subtopic of another large article. The article texts are of varying length ranging from one paragraph to multiple pages.

The articles have descriptors attached that were assigned by the professional indexers of the publisher. Usually, one, sometimes two, and exceptionally three subject descriptors are ascribed per article. The *descriptors* regard the broad subjects of the stories. They are used in matching article profiles with user's profiles in a routing task.

We use articles of the text classes CAR, INVESTMENTS, STOCK MARKET, CULINARY, FILM, COMPUTER SCIENCE, INTERNATIONAL, LITERATURE, MARKETING, MUSIC, POLITICS, SPORTS, TOURISM, and REAL ESTATE. The publisher has defined other classes, but the number of their members in the text corpus is too small to consider them in the experiments. A sample of 10 articles per class was manually analyzed, yielding the following characteristics of the texts of the classes.

1. The texts in the class CAR often describe new car and motor models and give their technical characteristics. They often exhibit a technical vocabulary. Sometimes, a text has the form of an index card that contains the technical details of the car.

2. The texts of the class INVESTMENTS bear upon different forms of investments (e.g., bonds, stocks, art, and real estate). They often overlap with texts in the classes STOCK MARKET and REAL ESTATE.
3. The texts of the class STOCK MARKET describe stock exchanges. They sometimes describe products of companies that offer stocks, which might result in a rich vocabulary.
4. CULINAIR describes culinary books, recipes, wines, restaurants, and cafés. The texts often exhibit a rich vocabulary due to descriptions of historical settings and locations of the places or to the variety of the ingredients of recipes.
5. The texts of the class FILM describe new films. Part of this description is rather technical, but another part of it gives a summary of the story of the film. The stories enrich the vocabulary of this class. Sometimes, an article contains a list of film titles and a short description of each film.
6. Texts of the class COMPUTER SCIENCE are technical in nature. They contain descriptions of companies and products.
7. The texts of the class INTERNATIONAL cover events outside Belgium. The main linguistic expressions that cue this class are often the names of foreign countries and important foreign personalities. The vocabulary in this text class is very rich. There is an overlap with the class POLITICS.
8. The texts of the class LITERATURE are commonly reviews of newly published books. As in the class FILM, the content is shortly described. But, there are some specific, technical cues such as references to the type of work, ISBN number, number of pages, and publisher.
9. The class MARKETING mostly contains detailed descriptions of products that are marketed. The products can be almost anything (e.g., animal food, insurance, and clothing). Sometimes, an article discusses multiple products. The vocabulary in these texts is very heterogeneous.
10. The class MUSIC contains texts about classical and modern music. The texts contain technical details and references to known composers.
11. The texts of the class POLITICS contain political events. They are rich in names of political personalities, parties, and organizations.
12. The texts of the class SPORTS describe sport events. Often, the technical vocabulary of a specific sport is present.
13. The class TOURISM contains travel stories and promotions of foreign places, cultures, and hotels. The texts are usually long with a rich vocabulary, However, an article in the form of an information scheme is possible.
14. The texts of the class REAL ESTATE describe the location, area, rent, price, and other characteristics of the real properties.

Two thirds of the articles are used for training and one third for testing (see below). The training corpus contains the following distribution of classes. There are about 300 examples of the class MARKETING and about 200 examples of the class CULINARY. There are about 150 examples of the classes INVESTMENTS, STOCK MARKET, TOURISM, and SPORTS, about 100 examples of the classes FILM, CAR, COMPUTER SCIENCE, MUSIC, and LITERATURE, and about 50 examples of the classes INTERNATIONAL, POLITICS, and REAL ESTATE. In the test corpus, the classes are present with about equal proportions.

A demonstrator is built in the programming language C on a Sun™ SPARC station 5 under Solaris® 2.5.1. It learns a text classifier from the training set and automatically assigns descriptors to new article texts with the learned classifier (Figure 1). It was agreed with the publisher that the system must simulate the manual process of assigning one or two descriptors to the articles, which reflect the main topics of the article.

3. METHODS: SUPERVISED LEARNING OF CLASSIFICATION PATTERNS

Because of the large and heterogeneous subject domain, we use *machine learning techniques* to acquire the textual patterns that imply the text classes (see chapter 5). The patterns are learned from a set of example texts (training set). An example text is represented as a set of features, which in some of the learning methods form a *vector of features*. Word and phrases are common features of texts and form the *multi-variate feature space* used in the classification problem. Selecting predictive features before training and transforming feature values to increase their predictive value are important. The learned classifier is used to assign subject descriptors or class labels to new, previously unseen texts. A new text is equally represented as a set of features.

The methods for classifying the magazine articles comprise an initial feature selection to identify and weigh important content terms in the texts, learning algorithms, assignment of subject descriptors, and example selection. Feature extraction that computes the strength of the relationship between the feature and the class is discussed as part of the learning method.

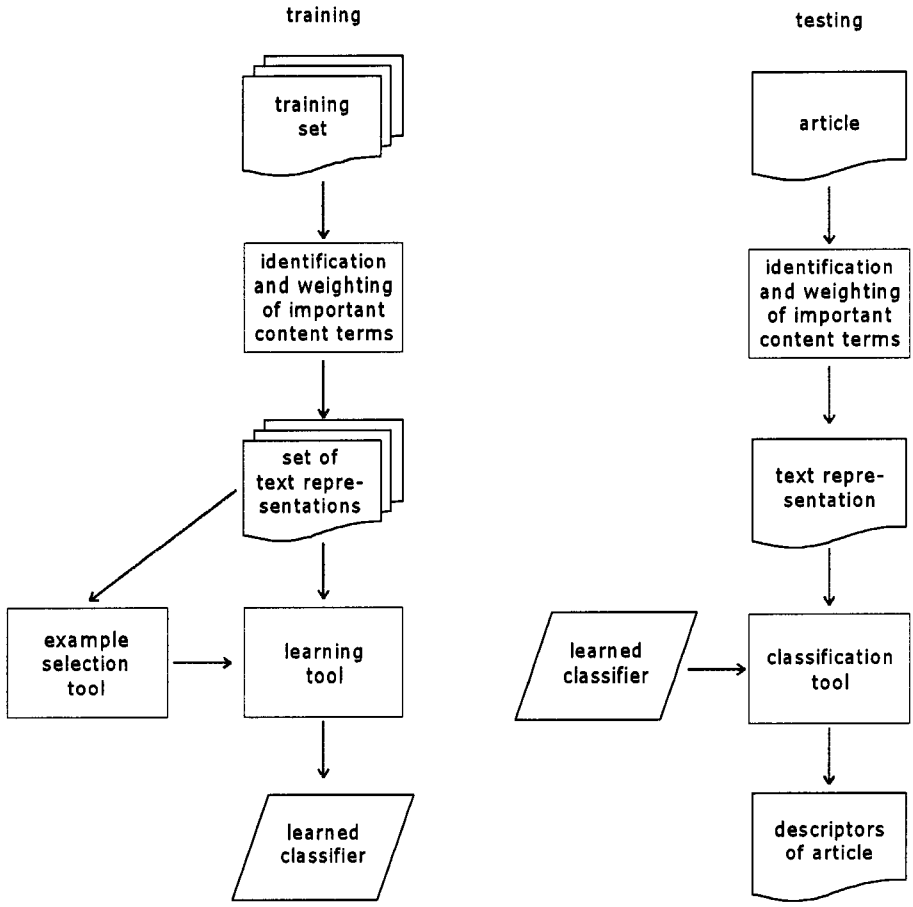


Figure I. Architecture of the text classifier.

3.1 Selection and Weighting of Important Content Terms

Words and proper name phrases are the salient features involved in classifying magazine articles. The articles contain many different words.

Because the text classes regard the main topics of the texts, it is important to identify important content terms in the text and to discard terms that do not bear upon content or treat only marginal topics. Identification of important content terms is part of processing the texts of the training and of the test corpus. Proper names are identified based on heuristic rules that take into account patterns of capitalized words and reoccurrence of these patterns

in the texts. Other content words are selected after elimination of stopwords. A *stoplist* of 879 non-content words is built based upon their syntactic classes. The stoplist contains function words such as articles, prepositions, auxiliary verbs, and others. Numbers are not accounted for. Currently, no form of stemming is used, except for the use of conjugated forms of auxiliary verbs in the stoplist. After removal of stopwords and numbers, we consider two different approaches for selecting important topic terms.

1. Words and proper names with a *high weight* are selected. The term frequency or the number of times a word or proper name occurs in the magazine article is a good measure for term importance. Because the articles are of varying length, the term frequency is normalized by its division by the maximum number of times a content term occurs in the text (chapter 5: formula (7)).
2. All content words and proper names *are selected from the* beginning of the article including the discourse segments of the *headline, lead, and the attribution of the article*. Magazine articles belong to the discourse type of written news stories. It is believed that in news stories the general topics come first to be followed by more specific information (van Dijk, 1985; Fairclough, 1995, p. 30). Content words and proper names are weighted with the term frequency.

A magazine article is represented with the words and proper names that are selected from it and their weights. It must be noted that noun phrases in Dutch often have the form of a single word that is the concatenation of the component words of the phrase. Consequently, some other noun phrases besides the proper names are part of the representations.

3.2 Learning Algorithms

We implemented several algorithms that belong to the group of statistical tools for pattern recognition. Among the algorithms, there are two variants of the *Bayesian independence classifier*, the *Rocchio* algorithm, and the χ^2 algorithm. Bayesian independence classification computes the posterior probability that a new, previously unseen text belongs to a certain text class given its features (here words and proper names). The Rocchio and the χ^2 algorithms generalize the positive and negative examples of each class into a category weight vector. The components of the weight vector are the text features (words and proper names) of the example texts. The weight of a feature indicates the strength of its relationship with the subject class. A category weight vector forms a kind of prototype example of the class. The

Bayesian and the Rocchio classifiers are discussed in detail in chapter 5 (formulas: (10), (15) and (17)). The choice of these algorithms is motivated by the fact that they have proven their usefulness in text categorization and are advantageous for comparisons.

The χ^2 (*chi-square*) test is a current technique for feature selection in relevance feedback and text categorization (e.g., Schütze, Hull, & Pedersen, 1995; Hull et al., 1997; Ng, Goh, & Low, 1997). The χ^2 *variable* is used to test how closely a set of *observed* frequencies corresponds to a set of *expected* frequencies. The observed frequencies are the number of texts relevant or non-relevant for the text class that contain the text feature (word or proper name) or not contain the feature (Table 1). The observed frequencies form an observed probability distribution. A useful probability distribution for the expected frequencies is that all expected frequencies of the presence of the feature (or of the absence of the feature) will be equal in texts relevant for the class and texts non-relevant for the class. The expected frequencies can be computed from the number of texts that contain the feature or do not contain the feature (Table 1).

Table 1. Contingency table of the χ^2 test relating to feature j and class C_k where N = the number of texts in the training set.

	Number of texts relevant for the class C_k	Number of texts non-relevant for the class C_k	
With feature j	N_{r+}	N_{n+}	$N_{r+} + N_{n+}$
Without feature j	N_{r-}	N_{n-}	$N_{r-} + N_{n-}$
	$N_{r+} + N_{r-}$	$N_{n+} + N_{n-}$	$N_{r+} + N_{n+} + N_{r-} + N_{n-} = N$

The χ^2 variable tests the hypothesis whether the observed and the expected frequencies are close enough to conclude that they come from the same probability distribution (*goodness-of-fit test)*. The formula for computing the χ^2*-variable* of a text feature j using the values of the above contingency table is:

$$\chi^2 = \frac{N(N_{r+}N_{n-} - N_{r-}N_{n+})^2}{(N_{r+} + N_{r-})(N_{n+} + N_{n-})(N_{r+} + N_{n+})(N_{r-} + N_{n-})} \quad (1)$$

When the resulting χ^2 *variable* is low, the fit of the observed and expected frequencies is good and hence the feature has no influence upon the text class. When the value is high, there is an association between the feature

and the class.[1] In text categorization this association is used to select features that are highly related to the text class.

We use the χ^2 variable in a different way. The relationship of a feature (word of proper name) is computed by applying the χ^2 test. Instead of selecting features with a high χ^2 value, the raw χ^2 values are used in the category weight vector. The above contingency table (Table 1) has 1 degree of freedom. Using the raw χ^2 values in a category weight vector and in similarity computation with the feature vector of a new text implicates that a term of the new text (word or proper name) that is related to the text class with a probability of more than 68% based on the training corpus has a positive effect upon assignment to the class (χ^2 value of more than 1 used in the inner product). High χ^2 values of 9 or more indicate a probability of close to 100% that the term is related to the text class. In our training set, excellent cues have χ^2 values of more than 200.

The above learning algorithms are implemented in such a way that, when new examples become available, the probabilities and category weight vectors can incrementally be learned without a computation from scratch.

Induction of rules and trees has a definite potential in text categorization, but is not considered here. Algorithms for rule induction screen the search space of possible hypotheses for the hypothesis that covers all (or most) of the positive examples and none (or fewest) of the negative examples. Because the magazine articles contain a large number of features, even after an initial feature selection, the number of possible hypotheses (e.g., Boolean combinations of text features, relations in first order logic) built from the positive and negative examples is enormous. The number of positive examples is too restricted to learn solely from them. Existing applications use greedy algorithms (e.g., Apté, Damerau, & Weiss, 1994), put constraints upon the complexity of the rules learned (e.g., Apté et al., 1994), or learn from very short texts (e.g., from titles: Cohen, 1995). Greedy algorithms consider only a subset of the hypotheses, whereby a rule is built by inclusion (general to specific) or exclusion (specific to general) of a single best feature without backtracking or with a limited backtracking.

3.3 Descriptor Assignment

For a new article to be classified by the Bayesian classifier, the probability of class membership is computed for each subject descriptor. The most probable descriptor is assigned. When a new article is classified with the Rocchio or χ^2 classifier, a *scoring function* computes the similarity between the feature vector of the new text to be classified and the weight vector of each class or category. We use the inner product of the vectors for

computing this similarity (Jones & Furnas, 1987). The subject descriptor of the category weight vector with highest similarity is assigned to the new article. In a variant implementation, a second descriptor is assigned when the probability of or the similarity with the second best class is less than 10% lower than the probability of or similarity with the best class.

3.4 Selecting Examples

Techniques of *supervised learning* are common in text categorization. Unsupervised learning techniques are less frequently used in text classification. In unsupervised learning the classes are not a-priori defined, but inferred from the data. The techniques include different forms of textual discovery: the discovery of groupings, relationships, and other patterns in text data. Clustering techniques have been developed to group texts based on words that they contain (e.g., Willett, 1988; Merkl, 1997). When selecting examples, we integrate an unsupervised learning component.

The technique of *zoning*, which is the selection of examples that are in close proximity of another example or concept description, has proven its effectiveness in relevance feedback (Schütze, Hull, & Pedersen, 1995; Singhal, Mitra, & Buckley, 1997). More specifically, a better query is learned from the documents that are judged relevant and from the *k*-nearest neighbors of the query that are judged non-relevant. Lam and Ho (1998) use this technique for training a text classifier. Because similarity between neighbors is based upon common terms in the examples, it is difficult to determine a single *k*-value that is effective both for text classes with a rich vocabulary and classes characterized by a much more restricted vocabulary. In Rocchio, negative examples are useful to give the noise terms of the positive examples a low weight in the category weight vector. Useful negative examples share many terms with the positive examples. The number of good negative neighbors may vary according to the richness of the vocabulary in a text class. We implemented an algorithm for zoning that selects negative examples based upon the similarity with the cluster of positive examples of the text class. Each negative example that does not decrease the average similarity between each pair of objects of the cluster of examples, when added to the cluster of positive examples, is considered in learning the class concept. Similarities are computed with the inner product applied upon the term vectors of the examples. Zoning is tested in combination with the Rocchio classifier. Because this technique learns from example texts that are similar in their word usage, zoning in combination with the χ^2 algorithm, which is based upon differences in word distributions, makes little sense.

Table 2. Results of the Bayesian independence classifier that considers the presence and absence of a term. Features are selected by considering the term frequency with length normalization and elimination of terms with low weights. 1 or 2 descriptors are assigned.

Class	Effectiveness measures		
	Recall	Precision	F-measure
CAR	0.844828	0.753846	0.796748
INVESTMENTS	0.420455	0.755102	0.540146
STOCK MARKET	0.586466	0.545455	0.565217
CULINARY	0.850746	0.890625	0.870229
FILM	0.897959	0.733333	0.807339
COMPUTER SCIENCE	0.847826	0.549296	0.666667
INTERNATIONAL	0.133333	0.500000	0.210526
LITERATURE	0.727273	0.533333	0.615385
MARKETING	0.484211	0.910891	0.632302
MUSIC	0.828125	0.757143	0.791045
POLITICS	0.250000	0.291667	0.269231
SPORTS	0.759259	0.745455	0.752294
TOURISM	0.628205	0.475728	0.541436
REAL ESTATE	0.423077	0.275000	0.333333
Average	0.620125	0.622633	0.599421

Another experiment regards the possibility of constructing *several category weight vectors* for a text class, when it represents a broad concept. Such a broad concept may be composed of many subconcepts and consequently be expressed in many different ways. Variant descriptions may overlap or exhibit a completely different word use. So, one class can be described with a few, different weight vectors, each of which represent the textual patterns of a subconcept. We propose an unsupervised learning method (*divisive clustering)* that in a natural way divides the positive instances of a class into subsets to be generalized with the Rocchio and χ^2 algorithms. For a broad text class, the cluster of positive examples is split in two and then in more clusters, if the new cluster structure better fits the positive examples. Better is defined in terms of a higher average fit of the examples to their cluster. The fit of an object to its cluster is defined in terms of the average similarity between an object and its cluster (this must be maximized) and the average similarity between the object and its second choice cluster (this must be minimized) (cf. chapter 8 formula (2)). Similarities are computed with the inner product applied upon the term vectors of the examples. At each step, the algorithm splits the cluster with "largest" diameter (diameter = smallest similarity between a pair of objects of the cluster) in two. The most isolated object of the cluster, i.e., the object with the smallest average similarity to the other objects, is removed from the cluster to form a new cluster (cf. Kaufman & Rousseeuw, 1990, p. 271 ff.). Other objects are moved to the new cluster, when they better fit the new

cluster. This is when the average similarity between the objects and their clusters in the two new clusters is higher than the average similarity between the objects in the old cluster. We limited the experiment of learning different category weight vectors to the text class SPORTS. SPORTS is a broad class which may have distinguished concept descriptions.

4. RESULTS AND DISCUSSION

We conducted a number of experiments aiming at comparing the initial feature selection methods, comparing the different algorithms for text categorization, and at possible improvements by example selection. The methods are tested upon a set of more than 930 new, previously unseen magazine articles that are manually classified by the professional indexers of the publisher.

4.1 Selection and Weighting of Important Content Terms

After elimination of stopwords and numbers and after normalization of upper case letters to lower case letters except for the proper names, the texts of the training set contain more than 60,000 different words. Identification of proper names and an initial selection of important content words and proper names in the training set results in a feature set of about 12,000 different words and proper names, when terms are selected based upon their weight in the complete text, and in a feature set of about 20,000 different words and proper names, when terms are selected from the beginning of the article texts. The texts of the articles of the test set are condensed in a like manner.

As we will see below, the categorization results are usually better when feature selection is based on the term frequency with length normalization with an elimination of low term weights than when features are selected only from the begin section of the article. When content terms are selected from topically important segments of the article such as the heading, lead and attribution parts, the set of terms still contains a number of common words.

4.2 Learning Algorithms and Descriptor Assignment

The effectiveness of automatic assignment of subject descriptors to the articles is computed by comparing the results with the assignment of subject descriptors to these texts by experts and by computing recall, precision, and F-measure values, which are common metrics for evaluating text

categorization (see chapter 5, p. 104-5). The comparisons are done automatically. We assign an equal importance to recall and precision in the F-measure ($\beta = 1$), punishing recall and precision values that are far apart. Recall, precision, and F-measure are ideally close to 1. The results are macro-averaged over categories.

The *Bayesian independence classifier* that accounts for the presence of a term in the new text to be classified (chapter 5 formula (15)) results in an average recall of 58%, average precision of 61%, and an average F-measure of 53%. The Bayesian independence classifier that accounts for the presence and absence of a term in the new text to be classified (chapter 5 formula (17)) results in an average recall of 62%, average precision of 62%, and an average F-measure of 60% (Table 2). It can be noted that classes with a low F-measure have the fewest positive examples in training (INTERNATIONAL, POLITICS, and REAL ESTATE). These are the results with an initial feature selection based on the term frequency with length normalization and with elimination of low term weights. We noted that the ratio of the probability that a text feature occurs given the positive examples of the class in the training corpus upon the probability that the text feature occurs in all the examples of the training corpus might be greater than one:

$$\frac{P(w_j = 1 | C_k = 1)}{P(w_j = 1)} \qquad \text{in some cases} > 1 \qquad (2)$$

where

$P(w_j = 1 | C_k = 1)$ = the probability that the feature w_j is present in a text of the example set that is relevant for the class C_k

$P(w_j = 1)$ = the probability that the feature w_j occurs in the complete training set.

This is the case when the feature is highly related to the positive examples of the class and not or only very slightly to the negative examples of the class. According to Lewis (1992b, p. 123), this situation only rarely occurs, however in our training corpus it occasionally occurs especially when the content word or phrase is an excellent cue of the text class. This means that the final value (chapter 5: formulas (15) and (17)) that is used for ranking and that is the product of the above probability proportions of the individual features can not be called a probability, and, more importantly, we can not speak of complete independence of features. On the contrary, some features become strongly dependent in the computation. Both variants of the Bayesian independence classifier that are implemented are sensible to this

situation. When the classifier also considers the absence of a term in the computations, a similar, but rare situation occurs when, the absence of a term is strongly related to the positive examples of the text class and the presence of the term is strongly related to the negative examples of the text class. For many text classes, this so-called Bayesian independence classifier produces satisfying results. We think this is because good cues in combination (e.g., “speed”, "horse power" for the class CAR) highly increase the final ranking value by the product of their individual erroneously called “probabilities” that are greater than 1. Given this inconsistency, we did not perform further testing with the Bayesian independence classifier.

The *Rocchio algorithm* results in an average recall of 64%, average precision of 57%, and an average F-measure of 54%, when positive and negative examples are equally weighted (Table 3). Slightly inferior results (average F-measure of 53%) are obtained when the relative importance of the negative examples is set to half of the positive examples. Again, classes with few positive examples score unsatisfactory low (INTERNATIONAL, POLITICS, REAL ESTATE). The class MARKETING trained upon 300 examples scores especially low. When training was based upon 150 examples, the F-measure for this class only rose slightly from 13% (Table 3) to 15%. The class MARKETING has a limited number of words and phrases that cue this class, while exhibiting a lot of noise features in the variant product descriptions.

Table 3. Results of the Rocchio algorithm with an equal weight of the positive and negative examples. Features are selected by considering the term frequency with length normalization and elimination of terms with low weights. 1 or 2 descriptors are assigned.

Class	Effectiveness measures		
	Recall	Precision	F-measure
CAR	0.724138	0.494118	0.587413
INVESTMENTS	0.448864	0.868132	0.591760
STOCK MARKET	0.864662	0.560976	0.680473
CULINARY	0.597015	0.563380	0.579710
FILM	0.877551	0.728814	0.796296
COMPUTER SCIENCE	0.630435	0.630435	0.630435
INTERNATIONAL	0.288889	0.722222	0.412698
LITERATURE	0.878788	0.537037	0.666667
MARKETING	0.068421	0.928571	0.127451
MUSIC	0.875000	0.658824	0.751678
POLITICS	0.607143	0.253731	0.357895
SPORTS	0.629630	0.459459	0.53125
TOURISM	0.730769	0.401408	0.518182
REAL ESTATE	0.692308	0.187500	0.295082
Average	0.636686	0.571043	0.537642

Table 4. Results of the χ^2 algorithm. Features are selected by considering the term frequency with length normalization and elimination of terms with low weights. 1 or 2 descriptors are assigned.

Class	Effectiveness measures		
	Recall	Precision	F-measure
CAR	0.827586	0.558140	0.666667
INVESTMENTS	0.789773	0.785311	0.787535
STOCK MARKET	0.481203	0.646465	0.551724
CULINARY	0.805970	0.613636	0.696774
FILM	1.000000	0.576471	0.731343
COMPUTER SCIENCE	0.673913	0.620000	0.645833
INTERNATIONAL	0.444444	0.689655	0.540541
LITERATURE	0.909091	0.526316	0.666667
MARKETING	0.457895	0.906250	0.608392
MUSIC	0.93750	0.759494	0.839161
POLITICS	0.607143	0.377778	0.465753
SPORTS	0.888889	0.623377	0.732824
TOURISM	0.743590	0.707317	0.725000
REAL ESTATE	0.653846	0.500000	0.566667
Average	0.730060	0.635015	0.658920

These are the results with an initial feature selection based on the term frequency with length normalization and with an elimination of low term weights. When the features are selected only from the begin section of the article and weighted by the term frequency, results of Rocchio with an equal weighting of positive and negative examples are worse having an average recall of 42%, average precision of 49%, and average F-measure of 41%. Selecting content words and proper name phrases from the begin section of an article results in more noise terms than selecting terms from the complete text based on high weights. For certain classes, the results are not necessarily inferior. This is the case for the class MARKETING: The F-measure is 13%, when important terms are selected from the complete text based upon their high weight. The F-measure is 28%, when only content terms of the beginning of the article are selected and weighted by the term frequency. In this class, the noisy terms of product descriptions usually appear further in the texts.

The χ^2 *algorithm* results in an average recall of 73%, average precision of 64%, and an average F-measure of 66% (Table 4). In this experiment, as in all the above experiments, one and occasionally two labels are assigned to the new articles. When we assign only one label under the same circumstances, the χ^2 algorithm results in an average recall of 69%, average precision of 68%, and an average F-measure of 66% (Table 5). It seems that the χ^2 algorithm is less sensitive to a varying number of positive examples

and in general less sensitive to noise terms. In the category weight vectors, noise terms have a very low weight compared to good cue terms.

The class MARKETING stills scores rather weak, but it scores significantly better than when applying the Rocchio classifier under the same circumstances. The χ^2 values in the category weight vector indicate that there are few good cue words and phrases for this class. Training with 150 positive examples yields an F-measure of the class MARKETING of 53% compared to an F-measure of 61% when training is based upon 300 positive examples (Table 4).

These are the results of the χ^2 classifier with an initial feature selection based on the term frequency with length normalization and on elimination of low term weights. When the features are selected only from the begin section of the article, results are inferior having an average recall of 63%, average precision of 57%, and average F-measure of 59%, when one or two descriptors are assigned, and an average recall of 60%, average precision of 60%, and average F-measure of 59%, when one descriptor is assigned.

In general the χ^2 algorithm scores much better than the other training methods tested. Figures 2 and 3 summarize the results of applying the Rocchio and χ^2 classifiers.

Table 5. Results of the χ^2 algorithm. Features are selected by considering the term frequency with length normalization and elimination of terms with low weights. 1 descriptor is assigned.

Class	Effectiveness measures		
	Recall	Precision	F-measure
CAR	0.775862	0.592105	0.671642
INVESTMENTS	0.721591	0.835526	0.774390
STOCK MARKET	0.368421	0.671233	0.475728
CULINARY	0.805970	0.683544	0.739726
FILM	1.000000	0.644737	0.784000
COMPUTER SCIENCE	0.608696	0.651163	0.629214
INTERNATIONAL	0.355556	0.666667	0.463768
LITERATURE	0.909091	0.588235	0.714286
MARKETING	0.389474	0.925000	0.548148
MUSIC	0.937500	0.800000	0.863309
POLITICS	0.571429	0.410256	0.477612
SPORTS	0.870370	0.643836	0.740158
TOURISM	0.705128	0.820895	0.758621
REAL ESTATE	0.615385	0.551724	0.581818
Average	0.688176	0.677494	0.658744

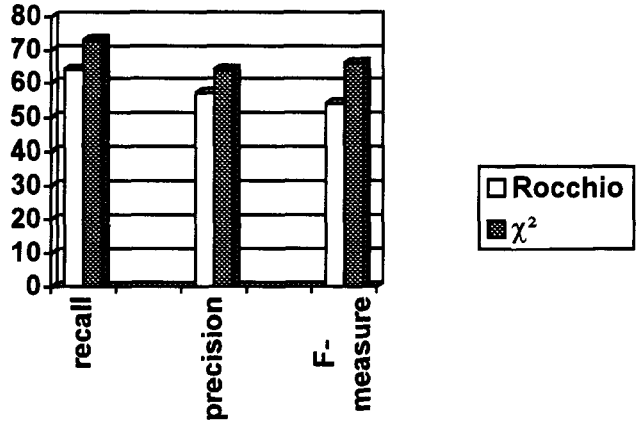


Figure 2. Comparisons of the average results in % of the Rocchio and χ^2 classifiers. Features are selected from the complete article by considering the term frequency with length normalization and elimination of terms with low weights. 1 or 2 descriptors are assigned.

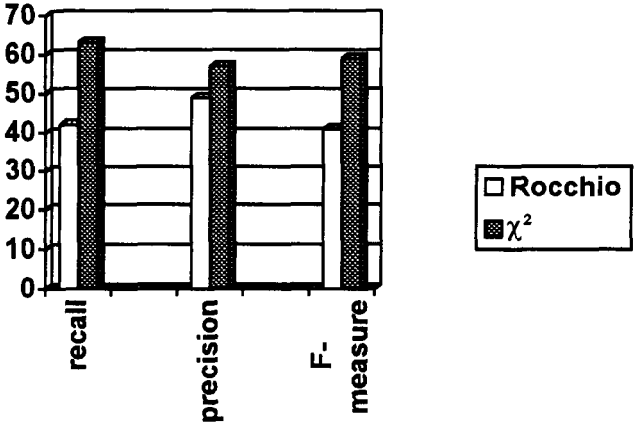


Figure 3. Comparisons of the average results in % of the Rocchio and χ^2 classifiers. Features are selected from the beginning of the article including the headline, lead, and attribution sections of the article and weighted with the term frequency. 1 or 2 descriptors are assigned.

4.3 Selecting Examples

Learning from the positive examples and a limited set of negative examples could not improve the results. This technique of *zoning* is applied with the Rocchio algorithm. It results in a very low average recall of 20%, an average precision of 61%, and an average F-measure of 22%. A possible explanation is that given the large variety of term usage in the articles, a selected set of negative examples does not help much in eliminating the noise in the limited set of positive examples. This is illustrated by the weight vector of the class MARKETING, which has many noise terms with a positive weight, and by the very bad F-measure of the class MARKETING (3%) in descriptor assignment. The class REAL ESTATE, which exhibits a low amount of noise terms, scores better with a selection of negative examples. Training with similar negative examples yields an F-measure of 43% instead of the 3 1% when the complete set of negative examples is used for training.

Divisive clustering of 80 positive examples of the class SPORTS results in five clusters, from which five category weight vectors are learned. Learning different category weight vectors of the class SPORTS could not improve the results. Recall is generally increased, but precision is decreased. When training with the Rocchio algorithm is based upon 80 positive examples of the class SPORTS, clustering of the examples yields a recall of 78% compared to 61% without clustering and a precision of 16% compared to 63% without clustering. When training with the χ^2 algorithm is based upon 80 positive examples of the class SPORTS, clustering of the examples yields a recall of 91% compared to 85% without clustering and a precision of 54% compared to 65% without clustering. The number of experiments is too limited to draw firm conclusions. Also, some changes to the cluster algorithms are needed. With a large number of text representations, which usually share few common terms, the cluster algorithms used in our experiments tend to form very large clusters in which the objects have a low average similarity. This relates to the way a cluster is split. A possible new cluster is built with an outsider object. We might investigate whether cluster algorithms that group objects based on the selection of representative objects are more suitable (see chapter 8). These algorithms result in a natural partitioning of the objects, but they are computationally more expensive.

4.4 Possible Improvements

We might consider the following procedures in order to further *improve the results* of the descriptor assignment. In case of selecting important content terms from the texts, stemming may ameliorate the results. For instance, inflectional morphemes can safely be removed (e.g., mapping of singular and plural forms of a noun to a single stem). In Dutch, nominal compounds are generally formed by concatenating two or more words to create a single word. Splitting compound nouns by applying a lexicon of words and word combination rules might also improve the results. Learning might be refined by supplementing examples to classes with a low number of examples (such as INTERNATIONAL, POLITICS, and REAL ESTATE). An alternative way of descriptor assignment is by setting a minimum threshold value in probability or similarity for class membership, which allows detecting when none of the classes apply. We might also test whether equal results are obtained with the χ^2 algorithm when terms with a low weight are removed from the category weight vectors, which would improve the efficiency of the comparisons with these weight vectors.

5. CONTRIBUTIONS OF THE RESEARCH

Because of the many advantages of assigning subject and classification codes to texts, automatic text categorization receives a large interest in current research. We refer to chapter 5 for a detailed overview of text categorization methods. Experiments in classification of Dutch texts are limited (an example is Ragas & Koster, 1998). Our research in classifying magazine articles complements the on-going investigations with the following contributions:

1. The χ^2 *classifier* scores much better than the other training methods tested and especially better than the Rocchio algorithm under the same circumstances. The χ^2 classifier is less sensitive to noise terms. In the category weight vectors, noise terms have a very low weight compared to good cue terms. The χ^2 test, which measures the fit between the observed and the expected frequencies of the content terms in the training corpus, is effective for identifying terms that are related to a specific text class. We use the raw χ^2 values in the category weight vector. At the same time of our research, using χ^2 values in the category weight vector was also suggested by Suzuki, Fukumoto, and Sekiguchi (1998). The benefit of this approach is explained and proved by our

experiments. The χ^2 values strongly distinguish terms that are highly related to a class from the ones that are related to a lesser degree. This is especially beneficial in ranking. Using the χ^2 values for ranking the new article texts according to their similarity to the category weight vectors is a novel technique with promising results. The results are significantly better than applying the classical statistical classification techniques to the same text corpus and to many results of text categorization in the literature. Good results are obtained despite a limited number of positive training examples. A limited number of positive examples is common in routing tasks. At any time new topics may be introduced in the document stream.

2. When subject descriptors deal with the *main topics*, a first *selection of features* in the texts that bear upon these topics is justified. A magazine article often contains many marginal topics that do not play a role in descriptor assignment. Term weighting by considering the term frequency divided by the maximum frequency that a term occurs in the text and selecting terms with a high weight is effective. This form of term weighting is also practical for identifying important proper names in the articles. These names complement the subject descriptors in a routing task. Discourse structure is important, when selecting content terms. But, the segments in which important content terms are located may differ from one text class to another.
3. We investigated whether the results of applying the Rocchio algorithm could be improved by training the classifier with a selection of negative examples that are similar to the positive ones (technique of *zoning*). Instead of relying upon a fixed number of *k* neighbors when selecting negative examples, examples are clustered in a natural way. It is shown that in case of a limited set of positive examples, learning from a selection of negative examples is advantageous for classes that exhibit a low amount of noise terms.

6. CONCLUSIONS

Successful systems that classify texts and assign subject or classification codes rely upon the words and phrases of the texts. In many text categorization situations the number of patterns is large to manually acquire. In this case, the classifier is trained upon example texts. We investigated three aspects of text classifiers when categorizing magazine articles with broad subject descriptors: feature selection, learning algorithms, and improvement of the quality of the learned classifier by selection and

grouping of the examples. Because the subject descriptors regard the broad topics of the texts, an initial feature selection that identifies the topic terms is important. Selecting important content words and proper names based upon the term frequency that is normalized by the maximum number a content term occurs in the text is effective. Adding knowledge of the discourse structure in the term selection process is useful for certain text classes. Given the limited number of positive examples and the high number of text features in the articles that belong to a variety of magazines, columns, and subject domains, the results of training a text classifier with the χ^2 algorithm are very satisfying.

---

[1] In case of a high χ^2 variable of a feature, it is assumed that many texts relevant for the text class to be learned contain the feature, while the feature is almost absent in texts non-relevant for the class. A high χ^2 variable may also refer to the situation in which the absence of the feature is typical for texts relevant for the class, while the feature occurs abundantly in texts that are non-relevant for the class. When there are a substantial number of text classes and when feature selection only considers important terms in the texts, this latter situation rarely occurs. A feature or term either occurs in all text classes or it is typical for one or a few classes.

SUMMARY AND FUTURE PROSPECTS

1. SUMMARY

The subject of this book, *"Automatic Indexing and Abstracting of Document Texts"*, is the automatic creation of content representations of documents. The documents contain texts written in natural language. This representation, which is called a *text representation*, is in the form of an indexing description or an abstract. It is used to facilitate the process of information or document selection in retrieval, browsing, and question-answering systems that operate upon large document collections. The text representation captures the topical content of the original text with varying degree of detail.

The first part of the book places *the subject in a broad context.*

Natural language text is an important means for communicating and storing information and there is now a plethora of textual databases. Because manual indexing and abstracting is no longer feasible, there is a pressing need for systems that automatically index and abstract document texts. Automatic indexing and abstracting is not new. Since the time of Luhn (1957, 1958) much research has been devoted to this topic. But, current indexing and abstracting systems are not up to the task. They often generate an incorrect and incomplete representation of a text's content. Such a crippled characterization of the content is the source of the many failings of current retrieval systems and one of the causes of the *information retrieval problem.* The inadequate characterization makes it impossible for information management systems to retrieve or select all documents (or

information) and only documents (or information) that are relevant to a specific need. Current text representations are often restricted to only certain terms that frequently occur in the text, or to all words from the beginning of the text, or to sentences that contain frequent terms. We assume that a representation that reflects the content in a semantically rich way will help solving the information retrieval problem in future systems. Recently, a number of alternative solutions have been proposed. They include full-text search, relevance feedback, information agents, and manual assignment of content attributes. We demonstrate that each of these solutions benefit from a more refined automatic characterization of the content of texts. However, it is presently not feasible to carry out complex and complete natural language processing of large and heterogeneous text collections. Nor is it always desirable in terms of efficiency. Such a process would require complete knowledge of the lexical, syntactic, semantic and discourse properties of the texts besides domain world and contextual knowledge. The real challenge is to find better text analysis methods that identify the main topics of a text as well as its subtopics with a minimum of reliance upon external knowledge.

The basic object of the research in this book is text and its content. It is essential to define the *attributes of text* that regard text content. We focus upon text written in Western European languages without going into detail about the language aspect of text. Texts come in many forms, which are commonly called text types or genres. The discipline of discourse analysis describes texts and explains their properties. The basic units of text are words and at a more detailed level of analysis letters, which are the basic symbols of written text, and phonemes, the basic sound units of spoken text. Letters and phonemes separately have no meaning, but combined into small meaning units called morphemes, they form the components of words. Letters and a number of marks form the character set of electronic texts. Words are combined into larger meaningful, linguistic units such as phrases, clauses, and sentences. At a micro level of description, discourse analysis concerns the vocabulary, syntax, and semantics of the individual sentences, clauses, and phrases. At a macro level discourse analysis goes beyond the sentence boundary and considers a text a complete grammatical unit. It includes the organization of text and the ways in which sentences are held together. Discourses, including texts, have important communicative goals and subgoals that are defined by their creator. To realize their goals, texts use a number of internal structures. The superstructure is the text-type dependent, formal organization of text in terms of the ordered parts it is composed of. The rhetorical structure involves the text-type independent relationships between sentences and clauses used to obtain a certain communicative effect. The thematic structure is the organization of the

topics and subtopics in the text. A user accesses the text with a specific focus of attention, which may touch on a portion of the creator's communicative goal. Another interesting aspect of discourse analysis studies how "surface" linguistic forms or phenomena signal the text structures and explains why these forms are chosen.

Indexing or abstracting the content of text results in a text representation, of which there are various forms. *Indexing* commonly extracts from or assigns to the text a set of single words or phrases that function as index terms of the text. These terms are commonly called natural language index terms. When the assigned words or phrases come from a fixed vocabulary, they are called controlled language index terms or descriptors. The controlled vocabulary can take the form of a thesaurus, a list of subject headings, or a broad classification scheme. Indexing with a controlled vocabulary is called text categorization. The index terms, besides reflecting content, can be used as access points or identifiers of the text in the document collection. Natural language index terms have the advantage of being expressive and flexible, of representing a variety of access points and perspectives of a text, and of easily representing new and complex concepts. Controlled language index terms, on the other hand, have the advantage of being unambiguous and of representing general access points to the classes of a document collection. Like indexing, *abstracting* also creates a reduced representation of the content of the text. Abstracts usually are in the form of a continuous, coherent text or of a profile that structures certain information from the text. There are many other formats, some of which blur distinction between indexing and abstracting. An abstract is highly valued as a condensed and comprehensible representation of a text's content.

Text representations as either indexing descriptions or abstracts may have two distinct functions in systems that make the information in document collections accessible. First, they indicate the content of the original text, In this capacity they are especially valuable for assessing the relevancy of the original text in systems that browse document collections and in those that filter the content of document collections. Second, indexing descriptions or abstracts can act as text surrogates while being informative of the content of the original text. In this form they are especially appreciated in systems that extract information from document collections (question-answering systems) and in systems that retrieve documents.

The second part of this book assesses the state of *existing techniques* for *indexing and abstracting* the content of text.

The majority of indexing techniques *selects natural language terms* from the texts. A frequently used process first identifies the individual words of a

text (lexical analysis) and removes words that do not bear upon content (stopwords). It then might conflate the words to their stem (stemming) and possibly recognizes phrases in the text. Finally, it weights words and phrases according to their importance in the text. Many of the techniques rely upon simple assumptions about distribution patterns of individual words. In some cases a limited amount of linguistic knowledge is employed bearing upon the vocabulary, syntax, and semantics of the individual sentences, clauses, and phrases, and in rare cases upon the syntax and semantics of the discourse. The linguistic knowledge is involved in stemming procedures, and in phrase recognition and normalization. The existing techniques for selection of natural language index terms from texts were originally developed to index heterogeneous document collections, which explains the rather shallow approach.

Automatic *assignment of controlled language index terms* to texts is based upon knowledge of typical patterns, such as words and their combinations, and of their relation with the concept represented by the index term. A first and common form of vocabulary control is to assign index terms as listed and described in a *thesaurus*. The thesaurus substitutes the individual words and phrases of the text with more uniform terms, hereby controlling synonymy and semantic ambiguity of the individual terms. A second important form of vocabulary control is the assignment of broad subject and classification codes, which is also called *text categorization*. The typical textual patterns (words and phrases) that imply the index term concepts and classes can be manually acquired and implemented respectively in a thesaurus and knowledge base. This is sometimes only realistic in a limited subject domain. Currently, there is a great interest to automating, at least partially, the knowledge acquisition step. This would not only reduce the cost of implementation, but also more importantly, automation would offer an opportunity to broaden the domain of the application. There are ongoing research efforts to automatically construct thesauri and acquire the knowledge of textual patterns involved in text categorization. But building thesauri automatically remains a very difficult task, although learning the classification patterns of broad text categories is somewhat easier.

The process of constructing a text classifier that generalizes from categorized example texts can build upon a long tradition of research in pattern recognition and of experiments in relevance feedback in retrieval. The problem is to correctly find the patterns in example texts that are associated with the subject heading or classification code. Statistical techniques of pattern recognition, learning of rules and trees, and training of neural nets are all based upon the principle that is, when a number of good

examples or a large number of examples are available, the desired patterns will be identified based upon re-occurring features, and noise will be neglected. However, there are an enormous number of features (words and phrases) in texts, many of which have no relevancy for classification. When the number of examples that are relevant for a class is restricted, which is often the case, prior knowledge must be included in the training process. This knowledge mainly concerns distribution and recognition of content-bearing words and phrases in texts.

The techniques of *automatic abstracting* focus mainly upon the analysis of the source text and the selection of salient information from it. The techniques fall into two classes. The ones that find their origins in natural language processing research rely heavily on symbolic knowledge and produce good quality abstracts. But they are often tied to a specific application. The knowledge is often semantic in nature reflecting the concepts of a specific subject domain. On the other hand, there are the more general techniques that have origins in information retrieval. They statistically process distribution patterns of words, but produce less accurate abstracts. There are now two promising research directions. One builds upon research in natural language processing. It shows an emerging interest in using discourse structures and their signaling surface cues for identifying relevant information in text to be included in the abstract. The other research direction builds upon work in information retrieval. It investigates automatic structuring of the text according to its topics based upon the distribution of its lexical items. Between these two approaches, there are the statistical techniques that learn the value of discourse patterns for a specific collection and its abstracts. They classify discourse parameters according to their importance based on example abstracts of example source texts.

The third part of the book considers *applications* that have been studied by the author.

Four applications elaborate on novel techniques for automatic indexing and abstracting and improve upon existing ones. This research is part of two projects: the SALOMON project and the Media On Line project. The techniques are implemented in demonstrators, which are designed and implemented by the author, and tested upon two large document collections of texts written in Dutch.

The two text collections are quite different in terms of discourse properties. The first, a corpus of legal texts contains the criminal cases from the Leuven court between 1992 and 1994. The cases have typical discourse structures that are explicitly signaled by the explicit ordering of the discourse segments and by cue words and phrases. The vocabulary exhibits much

variation, but important concepts in the texts, such as specific crimes, have consistent names. The second collection of texts is a corpus of articles from the magazines “Knack”, “Weekend Knack”, “Trends”, and “Cash!”. Many articles have the discourse structures typical of written news stories. But across magazines and columns, texts may exhibit other structures. In general the articles deal with a very large variety of topics as reflected in their heterogeneous vocabulary.

The SALOMON project constructs a system that automatically *summarizes Belgian criminal cases* in order to improve access to the large number of existing and future court decisions. SALOMON extracts relevant text units from the case text to form a case summary, which includes the following: the name of the court, the date of the decision, key paragraphs that describe the crimes committed, key paragraphs and concepts that express the essence of the opinion of the court, and references to essential foundations. Such a case profile makes it possible to rapidly define the relevance of the case. The summary may also be employed in text search. In a first abstracting step, SALOMON categorizes the cases and structures their texts into separate legally relevant and irrelevant components. A text grammar that mainly represents the discourse patterns of a criminal case is used to automatically determine the category of the case and its components. The grammar is implemented as a semantic network of frames. A parser is developed that parses the text based upon the text grammar. In this way, we are able to extract general data from the case and identify sections of text that are relevant for further abstracting. In the second step, SALOMON extracts informative text units of the alleged offences and of the court’s opinion using shallow statistical techniques. The research of this second step focuses on the development of novel techniques for automatic recognition of topical text paragraphs (or sentences). The application of cluster algorithms based on the selection of representative objects has potential for automatic theme recognition, text abstracting and text linking, even beyond the legal field. These techniques are employed to eliminate redundant material in the case texts, and to identify informative text paragraphs that are relevant to include in the case summary.

The SALOMON system succeeds in simulating part of the intellectual practice of abstracting. It identifies the topics of the text, recognizes its category, its structure and salient passages. It also deletes redundant and insignificant information, and selects thematically relevant text units and key terms. However, it is much more difficult to simulate the intellectual process involved in interpreting the text or to assign the meaning that the text has for a specific user at a specific moment in time. But, we believe that, when the machine succeeds in creating a more refined representation of a text’s

content, the contextual knowledge needed for its interpretation can be more readily related to the text's specific information content.

The two systems developed for the Media On Line project aim at a better representation of magazine articles for on-line selection. The first system creates *"highlight" abstracts* of the articles. A highlight abstract consists of clippings extracted from the article text and aims at attracting the reader's attention while he or she browses a database of article abstracts. Browsing this database is one way to select and buy relevant magazine articles on-line. The abstract must also deal with the main topics of the text. The system, which is a ported version of the system that categorizes and structures legal cases while using a different text grammar, is able to generate plausible abstracts from hard news and feature articles of the magazine "Knack". It employs the knowledge of discourse patterns that are typical of news stories. The research demonstrates that the typical discourse patterns can be implemented in a text grammar. The second technique developed for the Media On Line project *categorizes the articles using subject descriptors.* The descriptors, which represent broad topics (e.g., car, investments, and marketing), are used to effectively route articles to magazine subscribers who are interested in specific topics. In order to carry out categorization, a text classifier is developed that learns from categorized example texts. Different classical learning algorithms and one novel technique are then tested upon a corpus of articles belonging to different magazines and columns. We investigate three aspects of text classifiers: selection of features, learning algorithms, and improvement of the learned classifier by selection and grouping of the examples. The results of training a text classifier with the χ^2 algorithm were successful, given the limited number of positive examples and the high number of text features in the articles which come from a variety of magazines, columns, and subject domains.

In our applications, the techniques for *selecting natural language index terms* are used in building intermediate text representations. Weighted content terms are used in the term vectors when clustering the paragraphs of legal cases according to type of crime, or in the feature vectors of magazine articles to be used in text classification. We have explored two different ways of constructing a stoplist. The one for legal cases is made of high frequency terms in the document corpus. The one for the magazine articles is based upon syntactical classes that represent function words. In addition to term frequency and inverse document frequency weights, we have used two forms of length normalization of term weights. One, the cosine normalization is part of vector comparisons of the term vectors of paragraphs of the legal cases. The other form of length normalization normalizes the term weight by the maximum frequency of occurrence of a content term in

the text. This form is used when representing the texts of the magazine articles. The differences in approach are determined by the properties of the text corpora. The techniques employed in the corpus of magazine articles are better suited to heterogeneous texts with different styles and word usage. We have approached the process of *assigning controlled language index terms* or categorizing text in two different ways. We categorize the legal cases and their component segments by parsing the cases based upon a handcrafted knowledge base of typical text patterns. For building a text classifier that assigns subject descriptors to magazine articles, we rely upon machine learning techniques. Again the techniques employed on the corpus of magazine articles are better suited to heterogeneous text collections. When *abstracting texts*, we have pursued two different strategies. The first mainly relies upon knowledge of discourse structures and the subject domain. This strategy is useful in the first step of the abstracting of legal cases in order to recognize relevant and irrelevant text passages. It is also useful in creating highlight abstracts of magazine articles. The second strategy, which involves shallow statistical techniques, is useful for identifying the thematic structure in the offence and motivation parts in criminal cases. This technique is also beneficial for identifying representative paragraphs and sentences in these texts. The second strategy is used when the linguistic context of the information to be identified is not predictable.

The research discussed in this book demonstrates that progress can be made in automatic indexing and abstracting of a text's content without relying upon complete and complex natural language processing. That was the initial *hypothesis* of this publication. In the course of the book we have developed several subsidiary theses. In indexing and abstracting tasks it is useful to have knowledge of the discourse structures whether inherent or not to the text type or genre and of the surface linguistic cues that signal them. Texts, like sentences, have a kind of grammar: a set of implicit rules that creators and users have culturally acquired and assume when they work with text. These rules govern the selection and ordering of elements in discourse and make texts understandable. The structures of the text help in communicating the content of the texts and in focusing on information when using the texts. The research also demonstrates the need for an adequate and portable formalism for representing the discourse patterns. Moreover, it is found that some discourse patterns can be learned with unsupervised learning techniques, such as the topics of a text. Finally, the research demonstrates that the variant patterns of domain concepts can be learned with supervised learning techniques.

It is clear that the *contrast* between the *knowledge-based* techniques for automatic indexing and abstracting, with origins in natural language processing, and the *statistical techniques*, with origins in information retrieval research, becomes less pronounced. Statistical approaches help in acquiring the discourse patterns used in indexing and abstracting. They can be integrated with the knowledge-based ones with the precise nature of the integration depending on the specific application task.

Although this book focuses upon automatic indexing and abstracting of written text, many of the findings are also important for spoken text. Given the increasing use of spoken documents for communication and storage of information, the methods discussed can be of significant value. To use them it is only necessary that the text features employed can be automatically identified during speech recognition.

2. FUTURE PROSPECTS

Natural language is an important means of communicating and storing information. Automatic indexing and abstracting of the content of natural language texts will remain an important research topic. In the course of writing this book, it became clear that a great deal of interesting and challenging work remains to be done to bring automatic indexing and abstracting to maturity. It is not our purpose to examine all of the questions in detail, but briefly some fundamental prospects for future research are as follows:

1. When outlining the existing methods for indexing and abstracting document texts, it became clear that we are still far from the ideal indexing description or text abstract. More research is needed in text analysis and text representations to develop methods that allow representing the main content of a text and allowing us to hone in on specific information in a text. Progress can be made in many areas: 1) stemming; 2) the selection and normalization of phrases; 3) recognition of proper names and their semantic category; 4) recognition of word senses; 5) weighting of terms based on probability distributions of terms including the discrimination of insignificant words and the weighting of phrases; 6) thesaurus construction; 7) further integration of knowledge-based and statistical tools in text classification and summarization; 8) feature selection and extraction when recognizing patterns in texts; 9) supervised and unsupervised learning of classifications and discourse patterns; 10) theme recognition in texts including recognition of main topics and subtopics and topic relationships; 11) generalization of the

selected content in abstracts; 12) summarization of multiple texts; 13) and evaluation procedures. Progress will most probably be obtained by the coordinate efforts of natural language processing and information retrieval research.

2. Our research demonstrates that we need more studies of discourse and communication. This knowledge will be part of future text analysis and generation systems. Although we know that communication structures may evolve in time (Goody, 1986, p. 45 ff.), they nonetheless help us a great deal in finding information in text. Discourse studies involve the specific text types (whether or not they belong to specific professional settings) and more general studies of textual communication. Thematic structures of texts are especially interesting for automatic indexing and abstracting. While text grammars are promising for modeling text structures, more research is needed on their use in practical applications, as well as their integration in document grammars that model multimedia documents. Research is also called for on their compatibility with grammars used in text generation.
3. Finally, our research demonstrates that a basic knowledge of communication through text and natural language helps shape the techniques that automatically acquire discourse patterns from the texts of document collections or from individual texts. Automated acquisition of patterns is possible through generalizing patterns in a large number of texts or induction of patterns from a limited number of representative texts. Supervised and unsupervised learning is a promising research area and it may broaden the applicability of text analysis. The statistical techniques in their turn help us understand the communicative processes.
4. In a larger context, research efforts need to be directed towards adequate forms of text representations and their use in information retrieval and selection tools. Inference-based retrieval models such as the network and the logic-based models are probably the best methods for selecting documents or information from collections that have rich semantic representations of their texts. It is therefore likely that the next generation of artificial intelligence applications will be "text-based", rather than knowledge-based, deriving more power from stored text than from handcrafted rules (cf. Jacobs, 1992).

REFERENCES

Abbreviations:

IP&M: Information Processing & Management

JASIS: Journal of the American Society for Information Science

NIST SP: National Institute of Standards and Technology Special Publication

SIGIR Conference: Annual International ACM SIGIR Conference on Research and Development in Information Retrieval

Adamson, G.W., & Boreham, J. (1974). The use of an association measure based on character structure to identify semantically related pairs of words and document titles. *Information Storage and Retrieval*, 10, 253-260.

Agosti, M. (1996). An overview of hypertext. In M. Agosti & A. Smeaton (Eds.), *Information Retrieval and Hypertext* (pp. 27-47). Boston: Kluwer Academic Publishers.

Agosti, M., & Smeaton, A.F. (Eds.) (1996). *Information Retrieval and Hypertext*. Boston: Kluwer Academic Publishers.

Aha, D.W., Kibler, D., & Albert M.K. (1991). Instance-based learning algorithms. *Machine Learning*, 6, 37-66.

Aho, A.V., Sethi, R., & Ullman, J.D. (1986). *Compilers: Principles, Techniques, and Tools*. Reading, MA: Addison Wesley.

Allan, J. (1995). Relevance feedback with too much data. In E.A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th SIGIR Conference* (pp. 337-343). New York: ACM.

Allan, J., Callan, J., Croft, B., Ballesteros, L., Broglio, J., Xu, J., & Shu, H. (1997). INQUERY at TREC-5. In E.M. Voorhees & D.K. Harman (Eds.), *Information Technology: The Fifth Text REtrieval Conference (TREC-5)* (pp. 119-132). Gaithersburg, MD: NIST SP 500-238.

Allen, J. (1995). *Natural Language Understanding* (2nd ed.). Redwood City, CA: Benjamin/Cummings.

Allen, R.B. (1990). User models: theory, method, and practice. *International Journal of Man-Machine Studies*, 32, 511-543.

Alschuler, L. (1989). Hand-crafted hypertext - lessons from the ACM experiment. In E. Barrett (Ed.), *The Society of Text: Hypertext, Hypermedia and the Social Construction of Information* (pp. 343-361). Cambridge, MA: The MIT Press.

Anderberg, M.R. (1973). *Cluster Analysis for Applications*. New York: Academic Press.

Appelt, D.E., Hobbs, J.R., Bear, J., Israel, D., & Tyson, M. (1993). FASTUS: a finite-state processor for information extraction from real-world text. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence* (pp. 1172-1178). San Mateo, CA: Morgan Kaufmann.

Apté, C., Damerau, F., & Weiss, S.M. (1994). Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12 (3), 233-251.

Baker, L.D., & McCallum, A.K. (1998). Distributional Clustering of Words for Text Classification. In W.B. Croft, A. Moffat, C.J. van Rijsbergen, R. Wilkinson, & J. Zobel (Eds.), *Proceedings of the 21st SIGIR Conference* (pp. 96-103). New York: ACM.

Ballerini, J.-P., Büchel, M., Domenig, R., Knaus, D., Mateev, B., Mittendorf, E., Scháuble, P., Sheridan P., & Wechsler, M. (1997). SPIDER retrieval system at TREC-5. In E.M. Voorhees & D.K. Harman (Eds.), *Information Technology: The Fifth Text REtrieval Conference (TREC-5)* (pp. 217-228). Gaithersburg, MD: NIST SP 500-238.

Bánréti, Z. (1981). The topic of texts and the interpretation of texts. In J.S. Petöfi (Ed.), *Text vs Sentence: Continued* (pp. 43-57). Hamburg: Helmut Buske.

Barrett, E. (1989). Textual intervention, collaboration, and the online environment. In E. Barritt (Ed.), *The Society of Text. Hypertext, Hypermedia, and the Social Construction of Information* (pp. 305-321). Cambridge, MA: The MIT Press.

Barry, C.L. (1994). User-defined relevance criteria: an exploratory study. *JASIS,* 45 (3), 149-159.

Bateman, J.A. (1995). On the relationship between ontology construction and natural language: a socio-semiotic view. *International Journal Human-Computer Studies*, 43, 929-944.

Baxendale, P.B. (1958). Machine-made index for technical literature - an experiment. *IBM Journal of Research and Development,* 2 (4), 354-361.

Beghtol, C. (1986). Bibliographic classification theory and text linguistics: aboutness analysis, intertextuality and the cognitive act of classifying documents. *Journal of Documentation,* 42 (2), 84-113.

Belkin N.J., & Croft, W.B. (1992). Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, 35 (12), 29-48.

Bell, A. (1991). *The Language of News Media.* Oxford Blackwell.

Bernstein, L.M., & Williamson, R.E. (1984). Testing of a natural language retrieval system for a full text knowledge base. *JASIS*, 35 (4), 235-247.

Berrut, C., & Chiaramella, Y. (1989). Indexing medical reports in a multimedia environment: the RIME experimental approach. In *Proceedings of the Twelfth SIGIR Conference* (pp. 187-197). New York: ACM.

Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.

Black, J.B. (1981). The effects of reading purpose on memory for text. In J. Long & A. Baddeley (Eds.), *Attention and Performance LX* (pp. 347-361). Hillsdale, NJ: Lawrence Erlbaum.

Blair, D.C. (1990). *Language and Representation in Information Retrieval*. Amsterdam: Elsevier Science Publishers.

Blair, D.C., & Maron, M.E. (1985). An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Communications of the ACM,* 28 (3), 289-299.

Blair, D.C., & Maron, M.E. (1990). Full-text information retrieval: further analysis and clarification. *IP&M*, 26,437-447.

Blosseville, M.J., Hébrail, G., Monteil, M.G., & Pénot, N. (1992). Automatic document classification: natural language processing, statistical analysis, and expert system techniques used together. In N. Belkin, P. Ingwersen, & A.M. Pejtersen (Eds.), *Proceedings of the 15th SIGIR Conference* (pp. 51-57). New York: ACM.

Boguraev, B., & Briscoe, T. (Eds.) (1989). Computational Lexicography for Natural *Language Processing*. London: Longman.

Bonzi, S., & Liddy, E.D. (1989). The use of anaphoric resolution for document description in information retrieval. *IP&M*, 25 (4), 429-442.

Bookstein, A., Klein, S.T., & Raita, T. (1998). Clumping properties of content-bearing words. *JASIS*, 49 (2), 102-114.

Bookstein, A., & Swanson, D.R. (1974). Probabilistic models for automatic indexing. *JASIS*, 25 (5), 312-318.

Bookstein, A., & Swanson, D.R. (1975). A decision theoretic foundation for indexing. *JASIS*, 26 (1), 45-50.

Borko, H., & Bernick, M. (1963). Automatic document classification. *Journal of the ACM*, 10, 151-162.

Borko, H., & Bernier, C.L. (1975). *Abstracting Concepts and Methods*. New York: Academic Press.

Borko, H., & Bernier, C.L. (1978). *Indexing Concepts and Methods*. New York: Academic Press.

Boyce, B. (1982). Beyond topicality: a two stage view of relevance and the retrieval process. *IP&M,* 18 (3), 105-109.

Bradshaw, J.M. (Ed.) (1997). *Software Agents*. Menlo Park, CA: AAAI Press.

Brandow, R., Mitze, K., & Rau, L.F. (1995). Automatic condensation of electronic publications by sentence selection. *IP&M*, 31 (5), 675-685.

Branting, L.K., Lester, J.C. & Callaway, C.B. (1997). Automated drafting of self-explaining documents. In *Proceedings of the Sixth International Conference on Artificial Intelligence & Law* (pp. 72-81). New York: ACM.

Brookes, B.C. (1968). The measures of information retrieval effectiveness proposed by Swets. *Journal of Documentation*, 24, 41-54.

Brown, M.G., Foote, J.T., Jones, G.J.F., Sparck Jones, K., & Young, S.J. (1995). Automatic content-based retrieval of broadcast news. *Proceedings ACM Multimedia '95* (pp. 35-43). New York: ACM.

Brüninghaus, S., & Ashley, K.D. (1997). Finding factors: learning to classify case opinions under abstract fact categories. In *Proceedings of the Sixth International Conference on Artificial Intelligence and Law* (pp. 123-131). New York: ACM.

Bruza, P.D., & van der Weide, T.P. (1992). Stratified hypermedia structures for information disclosure. *The Computer Journal*, 35 (3), 208-220.

Buckley, C. (1993). The importance of proper weighting methods. Human Language Technology: *Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993* (pp. 349-352). San Francisco: Morgan Kaufmann.

Buckley, C., & Salton, G. (1995). Optimization of relevance feedback weights. In E.A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th SIGIR Conference* (pp. 351-357). New York: ACM.

Buckley, C., Salton, G., & Allan, J. (1992). Automatic retrieval with locality information using SMART. In D.K. Harman (Ed.), *The First Text REtrieval Conference (TREC-1)* (pp. 59-72). Washington: NIST SP 500-207.

Buckley, C., Salton, G., Allan, J., & Singhal, A. (1995). Automatic query expansion using SMART: TREC-3. In D.K. Harman (Ed.), *Overview of the Third Text REtrieval Conference (TREC-3)* (pp. 69-80). Gaithersburg, MD: NIST SP 500-225.

Burger, J.D., Aberdeen, J.S., & Palmer, D.D. (1997). Information retrieval and trainable natural language processing. In E. Voorhees & D.K. Harman (Eds.), *Information Technology: The Fifth Text REtrieval Conference (TREC-5)* (pp. 433-435). Gaithersburg, MD: NIST SP 500-238.

Burnett, M., Fisher, C., & Jones, K. (1996). InTEXT precision indexing in TREC-4. In D.K. Harman (Ed.), The Fourth Text REtrieval Conference (TREC-4) (pp. 287-294). Gaithersburg, MD: NIST SP 500-236.

Callan, J.P., & Lewis, D.D. (1994). The efficiency issues workshop report. In D.K. Harman (Ed.), *The Second Text REtrieval Conference (TREC-2)* (pp. 303-304). Gaithersburg, MD: NIST SP 500-215.

Carbonell, J. (1996). Digital librarians: beyond the digital book stack. *IEEE Expert*, June 1996, 11-13.

Chang, S.-K., & Leung, L. (1987). A knowledge-based message management system. *ACM Transactions on Office Information Systems*, 5 (3), 213-236.

Charniak, E. (1983). A parser with something for everyone. In M. King (Ed.), *Parsing Natural Lunguage* (pp. 117-149). London: Academic Press.

Chiaramella, Y. & Chevallet, J.P. (1992). About retrieval models and logic. *The Computer Journal*, 35 (3), 233-242.

Chiaramella, Y., & Kheirbek, A. (1996). An integrated model for hypermedia and information retrieval. In M. Agosti & A.F. Smeaton (Eds.). *Information Retrieval and Hypertext (*pp. 139-178). Boston: Kluwer.

Chiaramella, Y., & Nie, J. (1990). A retrieval model based on extended modal logic and its application to the RIME experimental approach. In J.-L. Vidick (Ed.), *Proceedings of the 13th SIGIR Conference* (pp. 25-43). New York: ACM.

Chinchor, N. (1992). MUC-4 Evaluation metrics. In Fourth Message Understanding Conference (MUC-4): *Proceedings of a Conference Held in McLean, Virginia June 16-18, 1992* (pp. 22-29). San Mateo, CA : Morgan Kaufmann.

Chinchor, N., Hirschman, L., & Lewis, D.D. (1993). Evaluating message understanding systems: an analysis of the third Message Understanding Conference (MUC-3). *Computational Linguistics*, 19 (3), 409-449.

Chomsky, N. (1975). *The Logical Structure of Linguistic Theory*. New York Plenum Press.

Ciravegna, F. (1995). Understanding messages in a diagnostic domain. *IP&M*, 31 (5), 687-701.

Cleveland, D.B., & Cleveland, A.D. (1990). *Introduction to Indexing and Abstracting* (2nd edition). Englewood, CO: Libraries Unlimited.

Cohen, J.D. (1995). Highlights: language- and domain-independent automatic indexing terms for abstracting. *JASIS*, 46 (3), 162-174.

Cohen, W.W. (1995). Text categorization and relational learning. In A. Prieditis & S. Russell (Eds.), *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 124-132). San Francisco: Morgan Kaufmann.

Collantes, L.Y. (1995). Degree of agreement in naming objects and concepts for information retrieval. *JASIS*, 46 (2), 116-132.

Conklin, J. (1987). Hypertext: an introduction and survey. *IEEE Computer*, 20 (9), 17-41.

Convey, J. (1992). *Online Information Retrieval. An Introductory Manual to Principles and Practice*. London: Library Association Publishing.

Cooper, W.S., Chen, A., & Gey, F.C. (1995). Experiments in the probabilistic retrieval of full text documents. In D.K. Harman (Ed.), *Overview of the Third Text REtrieval Conference (TREC-3)* (pp. 127-134). Gaithersburg, MD: NIST SP 500-225.

Coulmas F. (1989). *The Writing Systems of the World*. Oxford, UK: Basil Blackwell.

Coulthard, M. (Ed.) (1994). *Advances in Written TextAnalysis.* London: Routledge.

Cowie, J., & Lehnert, W. (1996). Information extraction. *Communications of the ACM,* 39 (1), 80-91.

Crawford, S.L., Fung, R.M., Appelbaum, L.A., & Tong, R.M. (1991). Classification trees for information retrieval. In L.A. Birnbaum & G.C. Collins (Eds.), *Machine Learning: Proceedings of the Eighth International Workshop (ML 91)* (pp. 245-249). San Mateo, CA: Morgan Kauhann.

Creecy, R.H., Masand, B.M., Smith, S.J., & Waltz, D.L. (1992). Trading MIPS and memory for knowledge engineering. *Communications of the ACM,* 35 (8), 48-64.

Cremmins, E.T. (1996). *The Art of Abstracting* (2nd edition). Arlington, VA: Information Resources Press.

Croft, W.B. (1987). Approaches to intelligent information retrieval. *IP&M,* 23 (4), 249-254.

Croft, W.B. (1993). Knowledge-based and statistical approaches to text retrieval. *IEEE EXPERT,* April 1993, 8-12.

Croft, W.B. (1995). Machine learning and information retrieval. In A. Prieditis & S. Russell (Eds.), *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 587). San Francisco: Morgan Kaufmann.

Croft, W.B., & Harper, D.J. (1979). Using probabilistic models of document retrieval without relevance information. *Journal of Documentation,* 35 (4), 285-295.

Croft, W.B., Krovetz, R., & Turtle, H. (1990). Interactive retrieval of complex documents. *IP&M,* 26 (5), 593-613.

Croft, W.B., & Turtle, H.R. (1992). Text retrieval and inference. In P.S. Jacobs (Ed.). *Text-based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval* (pp. 127-155). Hillsdale, NJ: Lawrence Erlbaum.

Croft, W.B., Turtle, H.R., & Lewis, D.D. (1991). The use of phrases and structured queries in information retrieval. In A. Bookstein, Y. Chiaramella, G. Salton, & V.V. Raghaven (Eds.), *Proceedings of the Fourteenth SIGIR Conference* (pp. 32-45). New York: ACM.

Cutting, D.R., Karger, D.R., Pedersen, J.O., & Tukey, J.W. (1992). Scatter/Gather: a cluster-based approach to browsing large document collections. In N.J. Belkin, P. Ingwersen, & A.M. Pejtersen (Eds.), *Proceedings of the Fifteenth SIGIR Conference* (pp. 318-329). New York: ACM.

Dahlgren, K. (1995). A linguistic ontology. *International Journal Human-Computer Studies,* 43, 809-818.

Damerau, F.J. (1993). Generating and evaluating domain-oriented multi-word terms for texts. *IP&M,* 29 (4), 433-447.

Danet, B. (1985). Legal discourse. In T.A. van Dijk (Ed.), *Handbook of Discourse Analysis* 1 (pp. 273-291). London: Academic Press.

De Beaugrande, R. (1985). Text linguistics in discourse studies. In T.A. van Dijk (Ed.), *Handbook of Discourse Analysis 1* (pp. 41-70). London: Academic Press.

De Beaugrande, R.-A., & Dressler, W.U. (1981). *Introduction to Text Linguistics*. London: Longman.

Dean, T., Allen, J., & Aloimonos, Y. (1995). *Artificial Intelligence: Theory and Practice*. Redwood City, CA: Benjamin/Cummings.

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., & Harshman, R. (1990). Indexing by latent semantic analysis. *JASIS,* 41 (6), 391-407.

DeJong, G. (1977). Skimmhg newspaper stories by computer. *In Proceedings of the 5th International Joint Conference on Artificial Intelligence* (p. 16). Cambridge, MA: William Kaufmann.

DeJong, G. (1982). An overview of the FRUMP system. In W.G. Lehnert & M.H. Ringle (Eds.), *Strategies for Natural Language Processing* (pp. 149-176). Hillsdale: Lawrence Erlbaum.

Del Favero, B., & Fung, R. (1994). Bayesian inference with node aggregation for information retrieval. In D.K. Harman (Ed), *The Second Text REtrieval Conference (TREC-2)* (pp. 151-161). Gaithersburg, MD: NIST SP 500-215.

Dennis, S.F. (1967). The design and testing of a fully automatic indexing-searching system for documents consisting of expository text. In G. Schecter (Ed.), *Information Retrieval: A Critical Review* (pp. 67-94). Washington D.C.: Thompson Book Company.

Dermatas, E., & Kokkinakis, G. (1995). Stochastic tagging of natural language texts. *Computational Linguistics*, 21 (2), 137-153.

Dillon, A. (1991). Readers' models of text structures: the case of academic articles. *International Journal Man-machine Studies*, 35, 913-925.

Dillon, M., & Gray, A.S. (1983). FASlT: a fully automatic syntactically based indexing system. *JASIS*, 34 (2), 99-108.

Dorfmüller-Karpusa, K. (1988). Temporal and aspectual relations as text-constitutive elements. In J.S. Petöfi (Ed.), *Text and Discourse Constitution: Empirical Aspects, Theoretical Approaches* (pp. 134-169). Berlin: Walter de Gruyter.

Duda, R.O., & Hart, P.E. (1973). *Pattern Classification and Scene Analysis*. New York: John Wiley & Sons.

Dumais, S.T. (1995). Latent Semantic Indexing (LSI): TREC-3 Report. In D.K. Harman (Ed.), *Overview of the Third Text REtrieval Conference (TREC-3)* (pp. 219-230). Gaithersburg, MD: NIST SP 500-225.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19 (1), 61-74.

Earl, L.L. (1970). Experiments in automatic extracting and indexing. *Information Storage and Retrieval*, 6 (6), 313-334.

Edmundson, H.P. (1 964). Problems in automatic abstracting. *Communications of the ACM*, 7 (4), 259-263.

Edmundson, H.P. (1969). New methods in automatic extracting. *Journal of the Association for Computing Machinery,* 16 (2), 264-285.

Edwards, J.S. (1991). *Building Knowledge-based Systems: Towards a Methodology*. London: Pitman Publishing.

Eirund, H., & Kreplin, K. (1988). Knowledge-based document classification supporting integrated document handling. In R.B. Allen (Ed.), *Conference on Office Information Systems* (pp. 189-196). New York: ACM.

Ellis, D., Furner-Hines, J, & Willett, P. (1994). On the creation of hypertext links in full-text documents: measurement of inter-linker consistency. *Journal of Documentation*, 50, 67-98.

Ellis, D., Furner, J., & Willett, P. (1996). On the creation of hypertext links in full-text documents: measurement of retrieval effectiveness. *JASIS*, 47 (4), 287-300.

Ellis, D.G. (1992). *From Language to Communication.* Hillsdale, NJ: Lawrence Erlbaum.

Endres-Niggemeyer, B. (1989). Content analysis - a special case of text comprehension. In S. Koskiala & R. Launo (Eds.), *Information * Knowledge * Evolution: Proceedings of the Forty-fourth FID Congress* (pp. 103-112). Amsterdam: North Holland.

Endres-Niggemeyer, B., & Neugebauer, E. (1998). Professional summarizing: no cognitive simulation without observation. *JASIS*, 49 (6), 486-506.

Evans, D.A., Ginther-Webster, K., Hart, M., Lefferts, R.G., & Monarch, I.A. (1991). Automatic indexing using selective NLP and first-order thesauri. In *RIAO 91 Conference Proceedings Intelligent Text and Image Handling* (pp. 624-643). Paris: C.I.D.-C.A.S.I.S.

Fagan, J.L. (1989). The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. *JASIS*, 40 (2), 115-132.

Fairclough, N. (1995). *Media Discourse*. London: Edward Arnold.

Faloutsos, C. (1992). Signature files. In W.B. Frakes & R. Baeza-Yates (Eds.), *Information Retrieval: Data Structures & Algorithms* (pp. 44-65). Englewood Cliffs, NJ: Prentice Hall.

Feng, C., & Michie, D. (1994). Machine learning of rules and trees. In D. Michie, D.J. Spiegelhalter, & C.C. Taylor (Eds.), *Machine Learning, Neural & Statistical Classification* (pp. 50-83). New York: Ellis Horwood.

Fidel, R. (1 994). User-centered indexing. *JASIS*, 45 (8), 572-576.

Fidel, R., & Efthimiadis, E.N. (1994). Terminological knowledge structure for intermediary expert systems. *IP&M,* 30 (1), 15-27.

Field, B.J. (1975). Towards automatic indexing: automatic assignment of controlled-language indexing and classification from free indexing. *Journal of Documentation*, 31 (4), 246-265.

Figge, U.L. (1979). Zur Konstitution einer Eigentlichen Textlinguistik. In J.S. Petöfi (Ed.), *Text vs Sentence: Basic Questions of Text Linguistics: First Part* (pp. 13-23). Hamburg: Helmut Buske Verlag.

Finch, S. (1995). Partial orders for document representation: a new methodology for combining document features. In E.A. Fox, P. Ingwersen, & R. Fidel (Eds.), *In Proceedings of the 18th SIGIR Conference* (pp. 264-272). New York: ACM.

Fitzpatrick, L., Dent, M., & Promhouse, G. (1997). Experiments with TREC using the Open Text Livelink Engine. In E. Voorhees & D.K. Harman (Eds.), *Information Technology: The Fifth Text REtrieval Conference (TREC-5)* (pp. 455-475). Gaithersburg, MD: NIST SP 500-238.

Fox, C. (1989). A stop list for general text. *SIGIR Forum*, 24 (1-2), 19-35.

Fox, C. (1992). Lexical analysis and stoplists. In W.B. Frakes & R. Baeza-Yates (Eds.), *Information Retrieval: Data Structures & Algorithms* (pp. 102-130). Englewood Cliffs, NJ: Prentice Hall.

Fox, E.A. (1980). Lexical relations: enhancing effectiveness of information retrieval systems. *ACM SIGIR Forum,* XV (3), 5-36.

Fox, E.A., Nutter, J.T., Ahlswede, T., Evens, M., & Markowitz, J. (1988). Building a large thesaurus for information retrieval. In *Second Conference on Applied Natural Language Processing* (pp. 101-108). Austin, TX: ACL.

Frakes, W.B. (1992). Introduction to information storage and retrieval systems. In W.B. Frakes & R. Baeza-Yates (Eds.), *Information Retrieval: Data Structures & Algorithms* (pp. 1-12). Englewood Cliffs, NJ: Prentice Hall.

Frakes, W.B., & Baeza-Yates, R. (1992). *Information Retrieval: Data Structures and Algorithms*. Englewood Cliffs, NJ: Prentice Hall.

Frants, V.I., Shapiro, J., & Voiskunskii, V.G. (1997). *Automated Information Retrieval: Theory and Methods*. San Diego: Academic Press.

Fries, P.H. (1994). On theme, rheme and discourse goals. In M. Coulthard (Ed.), *Advances in Written Text Analysis* (pp.229-249). London: Routledge

Froelich, T.J. (1994). Relevance reconsidered - towards an agenda for the 21st century: introduction to special topic issue on relevance research. *JASIS*, 45 (3), 124-133.

Fuhr, N. (1989). Models for retrieval with probabilistic indexing. *IP&M*, 25 (1), 55-72.

Fuhr, N. (1992). Probabilistic models in information retrieval. *The Computer Journal*, 35 (3), 243-255.

Fuhr, N., & Buckley, C. (1991). A probabilistic learning approach for document indexing. *ACM Transactions on Information Systems*, 9 (3), 223-248.

Fuhr, N., Gövert, N., & Rölleke, T. (1998). DOLORES: a system for logic-based retrieval of multimedia objects. In W.B. Croft, A. Moffat, & C.J. van Rijsbergen (Eds.), *Proceedings of the 21st SIGIR Conference* (pp. 257-265). New York: ACM.

Fuhr, N., & Knorz, G.E. (1984). Retrieval test evaluation of a rule based automatic indexing (AIR/PHYS). In C.J. van Rijsbergen (Ed.), *Research and Development in Information*

Retrieval: Proceedings of the Third Joint BCS and ACM Symposium (pp. 391-408). Cambridge, MA: Cambridge University Press.

Fuhr, N., & Pfeifer, U. (1994). Probabilistic information retrieval as a combination of abstraction, inductive learning, & probabilistic assumptions. *ACM Transactions on Information Systems,* 12 (1), 92-115.

Fum, D., Guida, G., & Tasso, C. (1985). Evaluating importance: a step towards text summarization. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* 2 (pp. 840-844). San Mateo, CA: Morgan Kaufmann.

Fung, R., & Del Favero, B. (1995). Applying Bayesian networks to information retrieval. *Communications of the ACM,* 38 (3), 42-48 and 57.

Furnas, G.W., Landauer, T.K., Gomez, L.M., & Dumais, S.T. (1987). The vocabulary problem in human-system communication. *Communications of the ACM,* 30 (11), 964-971.

García-Berrio, A., & Albaladejo Mayordomo, T. (1988). Compositional structure: macrostructures. In J.S. Petöfi (Ed.), *Text and Discourse Constitution: Empirical Aspects, Theoretical Approaches* (pp. 170-211). Berlin: Walter de Gruyter.

Gauch, S., & Smith, J.B. (1991). Search improvement via automatic query reformulation. *ACM Transactions on Information Systems,* 9 (3), 249-280.

Gelbart, D., & Smith, J.C. (1995). FLEXICON: An evaluation of a statistical ranking model adapted to intelligent legal text management. In *The Fourth International Conference on Artificial Intelligence and Law: Proceedings of the Conference* (pp. 142-149). New York: ACM.

Gey, F.C. (1994). Inferring probability of relevance using methods of logistic regression. In W.B. Croft & C.J. van Rijsbergen (Eds.), *Proceedings of the Seventeenth SIGIR Conference* (pp. 222-231). London: Springer.

Gilardoni, L., Prunotto, P., & Rocca, G. (1994). Hierarchical pattern matching for knowledge based news categorization. In *RIAO 94 Conference Proceedings Intelligent Multimedia Information Retrieval Systems and Management* (pp. 67-81). Paris: C.I.D.-C.A.S.I.S.

Goody, J. (1986). *The Logic of Writing and the Organization of Society.* Cambridge, UK: Cambridge University Press.

Gordon, M., & Pathak, P. (1999). Finding information on the World Wide Web: the retrieval effectiveness of search engines. *IP&M,* 35, 141-180.

Graesser, A.C., & Clark, L.F. (1985). *Structures and Procedures of Implicit Knowledge (Advances in Discourse Processes,* XVII). Nonvood, NJ: Ablex Publishing Corporation.

Green, R. (1995). Topical relevance relationships. I. Why topic matching fails. *JASIS,* 46 (9), 646-653.

Green, R., & Bean, C.A. (1995). Topical relevance relationships. II. An exploratory study and preliminary typology. *JASIS,* 46 (9), 654-662.

Greiff, W.R. (1998). A theory of term weighting based on exploratory data analysis. In W.B. Croft, A. Moffat, C.J. van Rijsbergen, R. Wilkinson, & J. Zobel (Eds.), *Proceedings of the 21st SIGIR Conference* (pp. 11-19). New York: ACM.

Griffiths, A., Luckhurst, H.C, & Willett, P. (1986). Using interdocument similarity information in document retrieval systems. *JASIS,* 37 (1), 3-11.

Grishman, R. (1986). *Computational Linguistics: An Introduction. Cambridge,* UK: Cambridge University Press.

Grishman, R., & Kittredge, R. (Eds.) (1986). *Analyzing Language in Restricted Domains: Sublanguage Description and Processing.* Hillsdale, NJ: Lawrence Erlbaum.

Grosz, B..J. (1981). Focusing and description in natural language dialogues. In A.K. Joshi, B.L. Webber, & I.A. Sag (Eds.), *Elements of Discourse Understanding* (pp. 84-105). Cambridge, UK: Cambridge University Press.

Grosz, B.J., & Sidner, C.L. (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12 (3), 175-204.

Gunnarsson, B.-L. (1997). Applied discourse analysis. In T.A. van Dijk (Ed.). *Discourse as Social Interaction* (*Discourse Studies: A Multidisciplinary Introduction* 2) (pp. 285-3 12). London: SAGE.

Guthrie, L., Pustejovsky, J., Wilks, Y., & Slator, B.M. (1996). The role of lexicons in natural language processing. *Communications of the ACM*, 39 (1), 63-72.

Haberlandt, K.F., & Graesser, A.C (1 985). Component processes in text comprehension and some of their interactions. *Journal of Experimental Psychology: General*, 114, 357-374.

Hafer, M.A., & Weiss, S.F. (1974). Word segmentation by letter successor varieties. *Information Storage and Retrieval*, 10, 371-385.

Hahn, U. (1989). Making understanders out of parsers: semantically driven parsing as a key concept for realistic text understanding applications. *International Journal of Intelligent Systems*, 4, 345-393.

Hahn, U. (1990). Topic parsing: accounting for text macro structures in full-text analysis. *IP&M*, 26 (1), 135-170.

Halicová, E., & Sgall, P. (1988). Topic and focus of a sentence and the patterning of a text. In J.S. Petöfi (Ed.), *Text and Discourse Constitution: Empirical Aspects, Theoretical Approaches* (pp. 70-96). Berlin: Walter de Gruyter.

Halliday, M.A.K. (1976). Theme and information in the English clause. In G.R. Kress & M.A.K. Halliday (Eds.), *Halliday: System and Function in Language* (pp. 174-188). London: Oxford University Press.

Halliday, M.A.K. (1989). *Spoken and Written Language.* Oxford: Oxford University Press.

Halliday, M.A.K., & Hasan, R. (1 976). *Cohesion in English.* London: Longman.

Hamill, K.A. & Zamora, A. (1980). The use of titles for automatic document classification. *JASIS*, 31 (5), 396-402.

Hand, D.J. (1 997). *Construction and Assessment of Classification* Rules. Chichester: John Wiley & Sons.

Hand, T.F (1997). A proposal for task based evaluation of text summarization systems. In *Proceedings of the ACL '97 Workshop on Intelligent Scalable Text Summarization (ISTS '97)* (pp. 31-38).

Harman, D. (1992a). Ranking algorithms. In W.B. Frakes & R. Baeza-Yates (Eds.), *Information Retrieval: Data Structures & Algorithms* (pp. 363-392). Englewood Cliffs, NJ Prentice Hall.

Harman, D. (1992b). Relevance feedback revisited. In N. Belkin, P. Ingwersen, & A.M. Pejtersen (Eds.), *Proceedings of the Fifteenth SIGIR Conference* (pp. 1-10). New York ACM.

Harman, D.K. (Ed.) (1993). *The First Text REtrieval Conference (TREC-1).* Gaithersburg, MD: NIST SP 500-207.

Harman, D.K. (Ed.) (1994). *The Second Text REtrievaI Conference (TREC-2).* Gaithersburg, MD: NIST SP 500-2 15.

Harman, D.K. (Ed.) (1995). *Overview of the Third Text REtrieval Conference (TREC-3).* Gaithersburg, MD: NIST SP 500-225.

Harman, D.K. (Ed.) (1996). *The Fourth Text REtrieval Conference (TREC-4).* Gaithersburg, MD: NIST SP 500-236.

Harter, S.P. (1975a). A probabilistic approach to automatic keyword indexing: Part I. On the distribution of specialty words in a technical literature. *JASIS*, 26 (4), 197-206.

Harter, S.P. (1975b). A probabilistic approach to automatic keyword indexing: Part II. An algorithm for probabilistic indexing. *JASIS*, 26 (5), 280-289.

Harter, S.P. (1986). *Online Information Retrieval: Concepts, Principles, and Techniques*. San Diego: Academic Press.

Hayes, P.J. (1992). Intelligent high-volume text processing using shallow, domain-specific techniques. In P. S. Jacobs (Ed.), *Text-based Intelligent Systems: Current Research and Practice in Infirmation Extraction and Retrieval* (pp. 227-241). Hillsdale: Lawrence Erlbaum.

Hayes, P. (1994). NameFinder: software that finds names in text. In *RIAO 94 Conference Proceedings Intelligent Multimedia Information Retrieval Systems and Management* (pp. 762-774). Paris: C.I.D.-C.A.S.I.S.

Hayes, P.J., & Weinstein, S.P. (1991). CONSTRUE/TIS: a system for content-based indexing of a database of news stories. In *2nd Annual Conference on Innovative Applications of Artificial Intelligence* (pp. 49-64). Menlo Park, CA: AAAl Press.

Hayes-Roth, B., & Hayes-Roth, F. (1977). Concept learning and the recognition and classification of exemplars. *Journal of Verbal Learning and Verbal Behavior*, 16, 321-338.

Hearst, M. (1994). Using categories to provide context for full-text retrieval results. In *RIAO 94 Conference Proceedings Intelligent Multimedia Information Retrieval Systems and Management* (pp. 115-129). Paris: C.I.D.-C.A.S.I.S.

Hearst, M.A. (1997). TextTiling: segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23 (1), 33-64.

Hearst, M.A., & Pedersen, J.O. (1996). Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In H.-P. Frei, D. Harman, P. Schaüble, & R. Wilkinson (Eds.), *Proceedings of the 19th SIGIR Conference* (pp. 76-84). New York: ACM.

Hearst, M.A., & Plaunt, C. (1993). Subtopic structuring for full-length document access. In R. Korfhage, E. Rasmussen, & P. Willett (Eds.), *Proceedings of the Sixteenth SIGIR Conference* (pp. 59-68). New York: ACM.

Henery, R.J. (1994). Methods for comparison. In D. Michie, D.J. Spiegelhalter, & C.C. Taylor (Eds.), *Machine Learning, Neural & Statistical Classfiication* (pp. 107-124). New York: Ellis Horwood.

Hersh, W.R., & Molnar, A. (1995). Towards new measures of information retrieval evaluation. In E.A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th SIGIR Conference* (pp. 164-170). New York: ACM.

Hertz, J., Krogh, A., Palmer, R.G. (1991). *Introduction to the theory of neural computation.* Redwood City, CA: Addison Wesley.

Hobbs, J.R. (1979). Coherence and coreference. *Cognitive Science*, 3,67-90.

Hobbs, J.R. (1993). Discourse. In *Human Language Technology. Proceedings of a Workshop Held at Plainsboro, New Jersey March 21-24* (p. 157-158). San Francisco: Morgan Kaufmann.

Hoch, R. (1994). Using IR techniques for text classification in document analysis. In W.B. Croft & C.J. van Rijsbergen (Eds.), *Proceedings of the Seventeenth SIGIR Conference* (pp. 31-40). London: Springer.

Horacek, H. & Zock, M. (Eds.) (1993). *New Concepts in Natural Language Generation: Planning, Realizations and Systems.* London: Pinter Publishers.

Hovy, E. (1993a). From interclausal relations to discourse structure - a long way behind, a long way ahead. In H. Horacek & M. Zock (Eds.), *New Concepts in Natural Language Generation: Planning, Realizations and Systems* (pp. 57-68). London: Pinter Publishers.

Hovy, E.H. (1993b). Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63, 341-385.

Hull, D. (1994). Improving text retrieval for the routing problem using Latent Semantic Indexing. In W.B. Croft & C.J. van Rijsbergen (Eds.), *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval* (pp. 282-289). London: Springer.

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

Hull, D.A., Grefenstette, G., Schütze, B.M., Gaussier, E., Schütze, H., & Pedersen, J.O. (1997). Xerox *TREC-5* site report: routing, filtering, NLP, and Spanish tracks. In E.M. Voorhees & D.K. Harman (Eds.), *Information Technology: The Fifth Text REtrieval Conference (TREC-5)* (pp. 167-180). Gaithersburg, MD: NIST SP 500-238.

Hutchins, J. (1987). Summarization: some problems and methods. In K.P. Jones (Ed.), *Meaning the Frontier of Informatics (Informatics* 9) (pp. 151-173). London: Aslib.

Hutchins, W.J. (1975). *Languages of Indexing and Classification: A Linguistic Study of Structures and Functions.* Stevenage, UK: Southgate House.

Hutchins, W.J. (1977). On the problem of aboutness in document analysis. *Journal of Informatics*, 1 (1), 17-35.

Hutchins, W.J. (1985). Information retrieval and text analysis. In T.A. van Dijk (Ed.), *Discourse and Communication: New Approaches to the Analysis of Mass Media Discourse and Communication* (pp. 106-125). Berlin: Walter de Gruyter.

Ide, E. (1971). New experiments in relevance feedback. In G. Salton (Ed.), *The SMART Retrieval System: Experiments in Automatic Document Processing* (pp. 337-354). Englewood Cliffs, NJ: Prentice-Hall.

Jacobs, P.S. (Ed.) (1 992). *Text-based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval.* Hillsdale, NJ: Lawrence Erlbaum.

Jacobs, P.S. (1993). Using statistical methods to improve knowledge-based news categorization. *IEEE expert*, April 1993, 13-23.

Jacobs, P.S., & Rau, L.F. (1990). "SCISOR": extracting information from on-line News. *Communications of the ACM*, 33 (11), 88-97.

Jacobs, P.S., & Rau, L.F. (1993). Innovations in text interpretation. *Artificial Intelligence*, 63, 143-191.

Jacquemin, C., & Royauté, J. (1994). Retrieving terms and their variants in a lexicalized unification-based framework. In W.B. Croft & C.J. van Rijsbergen (Eds.), *Proceedings of the Seventeenth SIGIR Conference* (pp. 132-141). London: Springer.

Jardine, N., & van Rijsbergen, C.J. (1971). The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7, 217-240.

Jing, Y., & Croft, W.B. (1994). An association thesaurus for information retrieval. In *RIAO 94 Conference Proceedings Intelligent Multimedia Information Retrieval Systems and Management* (pp. 146-160). Paris: C.I.D.-C.A.S.I.S.

Jonák, Z. (1984). Automatic indexing of full texts. *IP&M*, 20 (5/6), 619-627.

Jones, L.P., Gassie, E.W., & Radhakrishnan, S. (1990). INDEX: the statistical basis for an automatic conceptual phrase-indexing system. *JASIS*, 41, 87-97.

Jones, W.P., & Furnas, G.W. (1987). Pictures of relevance: a geometric analysis of similarity measures. *JASIS*, 38 (6), 420-442.

Kaufman, L., & Rousseeuw, P.J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis.* New York: John Wiley & Sons.

Kibler, D., & Aha, D.W. (1990). Comparing instance-averaging with instance-saving learning algorithms. In D.P. Benjamin (Ed.), *Change of Representation and Inductive Bias* (pp. 23 1-246). Boston: Kluwer Academic Publishers.

Kieras, D.E. (1985). Thematic processes in the comprehension of technical prose. In B.K. Britton & J.B. Black (Eds.), *Understanding Expository Text* (pp. 89-107). Hillsdale, NJ: Lawrence Erlbaum.

Kintsch, W., & van Dijk, T.A. (1978). Toward a model of text comprehension and production. *Psychological Review,* 85 (5), 363-394.

Kittredge, R., & Lehrberger, J. (Eds.) (1982). *Sublanguage: Studies of Language in Restricted Semantic Domains.* Berlin: Walter de Gruyter.

Koller, D., & Shoham, Y. (1996). Information agents: a new challenge for AI. *IEEE Expert*, June 1996, 8-10.

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

Kraaij, W. & Pohlmann, R. (1996). Viewing stemming as recall enhancement. In H.-P. Frei, D. Harman, P. Schaüble, & R. Wilkinson (Eds.), *Proceedings of the 19th SIGIR Conference* (pp. 4048). New York: ACM.

Krovetz, R. (1993). Viewing morphology as an inference process. In R. Korfhage, E. Rasmussen, & P. Willett (Eds.), *Proceedings of the Sixteenth SIGIR Conference* (pp. 191-202). New York: ACM.

Krovetz, R., & Croft, W. (1992). Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems,* 10 (2), 115- 141.

Krulee, G.K. (1991). *Computer Processing of Natural Language.* Englewood Cliffs, NJ: Prentice Hall.

Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. In E.A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th SIGIR Conference* (pp. 68-73). New York: ACM.

Lai, K.F., Lee, V.A.S., & Chew, J.P. (1996). Document routing by discriminant projection TREC-4. In D.K. Harman (Ed.), *The Fourth Text REtrieval Conference (TREC-4)* (pp. 449457). Gaithersburg, MD: NIST SP 500-236.

Lalmas, M. (1 998). Logical models in information retrieval: introduction and overview. *IP&M,* 34 (1), 19-33.

Lam, W., & Ho, C.Y. (1998). Using a generalized instance set for automatic text categorization. In W.B. Croft, A. Moffat, & C.J. van Rijsbergen (Eds.), *Proceedings of the 21st SIGIR Conference* (pp. 81-89). New York: ACM.

Lancaster, F.W. (1986). *Vocabulay Control for Information Retrieval* (2nd edition). Arlington, VA: Information Resources Press.

Lancaster, F.W. (1991). *Indexing and Abstracting in Theory and Practice.* London: The Library Association.

Lancaster, F.W., & Warner, A.J. (1993). *Information Retrieval Today.* Arlington, VA: Information Resources Press.

Lappin, S., & Leass, H.J. (1994). An algorithm for pronominal anaphora resolution. *Computational Linguistics,* 20 (4), 535-561.

Larkey, L.S., & Croft, W.B. (1996). Combining classifiers in text categorization. In H.-P. Frei, D. Harman, P. Schaüble, & R. Wilkinson (Eds.), *Proceedings of the 19th SIGIR Conference* (pp. 289-297). New York: ACM.

Lebart, L., Salem, A,, & Berry, L. (1997). *Exploring Textual Data.* Dordrecht, The Netherlands: Kluwer Academic Publishers.

Lee, J.H. (1995). Combining multiple evidence from different properties of weighting schemes, In E.A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th SIGIR Conference* (pp. 180-188). New York: ACM.

Lehnert, W.G. (1982). Plot units: a narrative summarization strategy. In W.G. Lehnert & M.H. Ringle (Eds.), *Strategies for Natural Language Processing* (pp. 375-412). Hillsdale, NJ: Lawrence Erlbaum.

Leung, C.-H., & Kan, W.-K. (1997). A statistical learning approach to automatic indexing of controlled index terms. *JASIS*, 48 (1), 55-66.

Lewis, D.D. (1992a). An evaluation of phrasal and clustered representations on a text categorization task. In N. Belkin, P. Ingwersen, & A.M. Pejtersen (Eds.), *Proceedings of the Fifteenth SIGIR Conference* (pp. 37-50). New York: ACM.

Lewis, D.D. (1992b). *Representation and Learning in Information Retrieval,* Ph.D. dissertation, University of Massachusetts.

Lewis, D.D. (1995). Evaluating and optimizing autonomous text classification systems. In E.A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th SIGIR Conference* (pp. 246-254). New York: ACM.

Lewis, D.D., Croft, W.B., & Bhandaru, N. (1989). Language-oriented information retrieval. International *Journal of Intelligent systems,* 4, 285-318.

Lewis, D.D., & Gale, W.A. (1994). A sequential algorithm for training text classifiers. In W.B. Croft & C.J. van Rijsbergen (Eds.), *Proceedings of the Seventeenth SIGIR* (pp. 3-11). London: Springer.

Lewis, D.D., Schapire, R.E., Callan, J.P., & Papka, R. (1996). Training algorithms for linear text classifiers. In H.-P. Frei, D. Harman, P. Schäuble, & R. Wilkinson (Eds.), *Proceedings of the 19th SIGIR Conference* (pp. 298-306). New York: ACM.

Lewis, D.D., & Sparck Jones, K. (1996). Natural language processing for information retrieval. *Communications of the ACM*, 39 (1), 92-101.

Liddy, E. (1990). Anaphora in natural language processing and information retrieval. *IP&M,* 26 (1), 39-52.

Liddy, E.D., McVaeny, K.A., Paik, W., Yu, E., & McKenna, M. (1993). Development, implementation and testing of a discourse model for newspaper texts. In *Human Language Technology. Proceedings of a Workshop Held at Plainsboro, New Jersey March 21-24* (pp. 159-164). San Francisco: Morgan Kaufmann.

Liddy, E.D., & Myaeng, S.H. (1993). DR-LINK'S: linguistic-conceptual approach to document detection. In D.K. Harman (Ed.), *The First Text REtrieval Conference (TREC-1)* (pp. 113-129). Gaithersburg, MD: NIST SP 500-207.

Liddy, E.D., & Paik, W. (1993). Document filtering using semantic information from a machine readable dictionary: preliminary test results. *Proceedings of the ACL Workshop on Very Large Corpora* (15 pp.).

Liddy, E.D., Paik, W., & Yu, E.S. (1994). Text categorization for multiple users based on semantic features from a machine-readable dictionary. *ACM Transactions on Information Systems,* 12 (3), 278-295.

Losee, R.M. (1988). Parameter estimation for probabilistic document-retrieval models. *JASIS,* 39 (1), 8-16.

Lovins, J.B. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics,* 11 (1-2), 22-31.

Lucarella, D., & Zanzi, A. (1996). Information modelling and retrieval in hypermedia systems. In M. Agosti & A.F. Smeaton (Eds.), *Information Retrieval and Hypertext* (pp. 121-138). Boston: Kluwer Academic Publishers.

Luhn, H.P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1 (4), 309-317.

Luhn, H.P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2 (2), 159-165.

Maeda, T., Momouchi, Y., & Sawamura, H. (1980). An automatic method for extracting significant phrases in scientific or technical documents. *IP&M*, 16,119-127.

Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM*, 37 (7), 31-40. Also published in J.M. Bradshaw (Ed.) (1 997), *Software Agents* (pp. 145-164). Menlo Park, CA: AAAI Press.

Mani, I., & Bloedorn, E. (1997). Multi-document summarization by graph search and matching. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference* (pp. 622-628). Menlo Park, CA: The MIT Press.

Mani, I., & Bloedorn, E. (1998). Machine learning of generic and user-focused summarization. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference* (pp. 821-826). Menlo Park, CA: The MIT Press.

Mani, I., & MacMillan, T.K. (1996). Identifying proper names in newswire text. In B. Boguraev & J. Pustejovsky (Eds.), *Corpus Processing for Lexical Acquisition* (pp. 41-59). Cambridge, MA: The MlT Press.

Mann, W.C., Matthiessen, C.M.I.M., & Thompson, S.A. (1992). Rhetorical Structure Theory and text analysis. In W.C. Mann & S.A. Thompson (Eds.), *Discourse Description: Diverse Linguistic Analyses of a Fund-raising Text* (pp. 39-78). Amsterdam: John Benjamins.

Margulis, E.L. (1992). N-Poisson document modelling. In N. Belkin, P. Ingwersen, & A.M. Pejtersen (Eds.), *Proceedings of the Fifteenth SIGIR Conference* (pp. 177-189). New York: ACM.

Margulis, E.L. (1993). Modelling documents with Multiple Poisson distributions. *IP&M*, 29 (2), 215-227.

Maron, M. (1961). Automatic indexing: an experimental inquiry. *Journal of the ACM*, 8,404-417.

Maron, M.E. (1977). On indexing, retrieval and the meaning of about. *JASIS*, 28, 38-43.

Maron, M.E., & J.L. Kuhns (1960). On relevance, probabilistic indexing and information retrieval. *Journal of the Association for Computing Machinery*, 7 (3), 216-244.

Masand, B., Linoff, G., & Waltz, D. (1992). Classifying news stories using Memory Based Reasoning. In *Proceedings of the Fifteenth SIGIR Conference* (pp. 59-65). New York: ACM.

Mathis, B.A., Rush, J.E., & Young, C.E. (1973). Improvement of automatic abstracts by the use of structural analysis. *JASIS*, 24 (2), 101-109.

Mauldin, M.L. (1991). Retrieval performance in FERRET: a conceptual information retrieval system. In A. Bookstein, Y. Chiaramella, G. Salton, & V.V. Raghaven (Eds.), *Proceedings of the Fourteenth SIGIR Conference* (pp. 347-355). New York: ACM.

Mc Cune, B.P., Tong, R.M., Dean, J.S., & Shapiro, D.G. (1985). RUBRIC: a system for rule-based information retrieval. *IEEE Transactions on Software Engineering*, 11 (9), 939-945.

McArthur, T. (1987). Representing knowledge for human consumption. In K.P. Jones (Ed.), *Meaning: The Frontier of Informatics (Informatics 9)* (pp. 9-19). London: Aslib.

McDonald, D.D. (1992). Robust partial-parsing through incremental, multi-algorithm processing. In P.S. Jacobs (Ed.). *Text-based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval* (pp. 83-99). Hillsdale, NJ: Lawrence Erlbaum.

McDonald, D.D. (1993). Does natural language generation start from a specification? In H. Horacek & M. Zock (Eds.), *New Concepts in Natural Language Generation* (pp. 275-278). London: Pinta Publishers.

McDonald, D.D. (1996). Internal and external evidence in the identification and semantic categorization of proper names. In B. Boguraev & J. Pustejovsky (Eds.), *Corpus Processing for Lexical Acquisition* (pp. 21-39). Cambridge, MA: The MIT Press.

McKeown, K., & Radev, D.R. (1995). Generating summaries of multiple news articles. In E.A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th SIGIR Conference* (pp. 74-82). New York: ACM.

McKeown, K., Robin, J., & Kukich, K. (1995). Generating concise natural language summaries. *IP&M*, 31 (9,703-733.

McKinin, E.J., Sievert, E., Johnson, E.D. & Mitchell, J.A. (1991). Medline/Full-text research project. *JASIS*, 42 (4), 297-307.

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

Merkl, D. (1997). Exploration of text collections with hierarchical feature maps. In N.J. Belkin, A.D. Narasimhalu, & P. Willett (Eds.), *Proceedings of the 20th SIGIR Conference* (pp. 186-195). New York: ACM.

Metzler, D.P., & Haas, S.W. (1989). The constituent object parser: syntactic structure matching for information retrieval. *ACM Transactions on Information Systems*, 7 (3), 292-316.

Meyer, B.J.F. (1985). Prose analysis: purposes, procedures, and problems. In B.K. Britton & J.B. Black (Eds.), *Understanding Expository Text* (pp. 11-64). Hillsdale, NJ: Lawrence Erlbaum.

Michie, D., Spiegelhalter, D.J., & Taylor, C.C. (Eds.) (1994). *Machine Learning, Neural and Statistical Classification.* New York: Ellis Horwood.

Miike, S., Itoh, E., Ono, K., & Sumita, K. (1994). A full-text retrieval system with a dynamic abstract generation function. In W.B. Croft & C.J. van Rijsbergen (Eds.), *Proceedings of the Seventeenth SIGIR Conference* (pp. 152-161). London: Springer.

Miller, G.A. (Ed.) (1990). Special issue: WordNet: an on-line lexical database. *International Journal of Lexicography*, 3 (4).

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

Miller, U. (1997). Thesaurus construction: problems and their roots. *IP&M*, 33 (4), 481-493.

Mitchell, T.M. (1977). Version spaces: a candidate elimination approach to rule learning. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence* (pp. 305-310). Cambridge, MA: William Kaufmann.

Moens, M.-F., & Dumortier, J. (1998). Automatic abstracting of magazine articles: the creation of 'highlight' abstracts. In W.B. Croft, A. Moffat, C.J. van Rijsbergen, R. Wilkinson, & J. Zobel (Eds.), *Proceedings of the 21st SIGIR Conference* (pp. 359-360). New York: ACM.

Moens, M.-F., & Uyttendaele, C. (1997). Automatic structuring and categorization as a first step in summarizing legal cases. *IP&M*, 33 (6), 727-737.

Moens, M.-F., Uyttendaele, C., & Dumortier, J. (1997). Abstracting of legal cases: the SALOMON experience. In *Proceedings of the Sixth International Conference on Artificial Intelligence & Law* (pp. 114-122). New York: ACM.

Moens, M.-F., Uyttendaele, C., & Dumortier, J. (1999a). Abstracting of legal cases: the potential of clustering based on the selection of representative objects. *JASIS*, 50 (2), 151-161.

Moens, M.-F., Uyttendaele, C., & Dumortier, J. (1999b). Information extraction from legal texts: the potential of discourse analysis. *International Journal of Human-Computer Studies*, 51, 1155-1171.

MUC-4 (1992). *Proceedings of the Fourth Message Understanding Conference (MUC-4).* San Mateo, CA: Morgan Kaufmann.

Ng, H.T., Goh, W.B., & Low, K.L. (1997). Feature selection, perceptron learning, and a usability case study for text categorization. In N.J. Belkin, A.D. Narasimhalu, & P. Willett (Eds.), *Proceedings of the 20th SIGIR Conference* (pp. 67-73). New York: ACM.

Ng, K.B., Loewenstem, D., Basu, C., Hirsh, H., & Kantor, P.B. (1997). Data fusion of machine-learning methods for the TREC5 routing task (and other work). In E.M. Voorhees & D.K. Harman (Eds.), *Information Technology: The Fifth Text REtrieval Conference (TREC-5)* (pp. 477-487). Gaithersburg, MD: NIST SP 500-238.

Nie, J. (1989). An information retrieval model based on modal logic. *IP&M*, 25 (5), 477-494.

Nie, J.-Y. (1992). Towards a probabilistic modal logic for semantic based information retrieval. In N.J. Belkin, P. Ingwersen, & A.M. Pejtersen (Eds.), Proceedings of the Fifteenth SIGIR Conference (pp. 140-151). New York: ACM.

Nielsen, J. (1995). *Multimedia and Hypertext: The Internet and Beyond.* Boston: AP Professional.

Nilsson, N.J. (1990). *The Mathematical Foundations of Learning Machines.* San Mateo, CA: Morgan Kaufmann.

Noreault, T., McGill, M., & Koll, M.B. (1981). A performance evaluation of simlarity measures, document term weighting schemes and representations in a Boolean environment. In R.N. Oddy, S.E. Robertson, C.J. van Rijsbergen, & P.W. Williams (Eds.), *Information Retrieval Research* (pp. 57-76). London: Butterworth & Co.

Oard, D.W. (1997). Alignment of Spanish and English TREC topic descriptions. In E.M. Voorhees & D.K. Harman (Eds.), *Information Technology: The Fifth Text REtrieval Conference* (TREC-5) (pp. 547-553). Gaithersburg, MD: NIST SP 500-238.

Paice, C.D. (1981). The automatic generation of literature abstracts: an approach based on the identification of self-indicating phrases. In R.N. Oddy, S.E. Robertson, C.J. van Rijsbergen, & P.W. Williams (Eds.), *Information Retrieval Research* (pp. 172-191). London: Butterworth & Co.

Paice, C.D. (1990). Constructing literature abstracts by computer: techniques and prospects. *IP&M*, 26 (1), 171-186.

Paice, C.D. (1991). The rhetorical structure of expository text. In K.P. Jones (Ed.), *The Structuring of Information (Informatics 11)* (pp. 1-25). London: Aslib.

Paice, C.D., & Jones, P.A. (1993). The identification of important concepts in highly structured technical papers. In R. Korfhage, E. Rasmussen, & P. Willett (Eds.), *Proceedings of the Sixteenth SIGIR Conference* (pp. 69-78). New York: ACM.

Paik, W., Liddy, E.D., Yu, E., & McKenna, M. (1993). Interpretation of proper nouns for information retrieval. *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993* (pp. 309-313). San Francisco: Morgan Kaufmann.

Paik, W., Liddy, E.D., Yu, E., & McKenna, M. (1996). Categorizing and standardizing proper nouns for efficient information retrieval. In B. Boguraev & J. Pustejovsky (Eds.), *Corpus Processing for Lexical Acquisition* (pp. 61-73). Cambridge, MA: The MIT Press.

Pao, M.L. (1987). *Concepts of Information Retrieval.* Englewood, CO: Libraries Unlimited.

Peat, H.J., & Willett, P. (1991). The limitations of term co-occurrence data for query expansion in document retrieval systems. *JASIS*, 42 (5), 378-383.

Pfeifer, U., Poersch, T., & Fuhr, N. (1996). Retrieval effectiveness of proper name search methods. *IP&M*, 32 (6), 667-679.

Pinto Molina, M. (1995). Documentary abstracting: toward a methodological model. *JASIS*, 46 (3), 225-234.

Porter, M.F. (1980). An algorithm for suffix stripping. *Program*, 14 (3), 130-137.

Pozzi, S., & Celentano, A. (1993). Knowledge-based document filing. *IEEE Expert*, 8 (5), 34-45.

Prikhod'ko, S.M., & Skorokhod'ko, E.F. (1982). Automatic abstracting from analysis of links between phrases. *Nauchno-Tekhnicheskaya Informatsiya, Seriya* 2, 16 (1). 27-32.

Quinlan, J.R. (1986). The effect of noise on concept learning. In S.R. Michalski, J.G. Carbonell, & T.M. Mitchell (Eds.), *Machine Learning. An Artificial Intelligence Approach II* (pp. 149-166). Los Altos, CA: Morgan Kaufmann.

Quinlan, J.R. (1990). Learning logical definitions from relations. *Machine Learning*, 5 (3), 239-266.

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann.

Ragas, H., & Koster, C.H.A. (1998). Four text classification algorithms compared on a Dutch corpus. In W.B. Croft, A. Moffat, & C.J. van Rijsbergen (Eds.), *Proceedings of the 21st SIGIR Conference* (pp. 369-370). New York: ACM.

Raghaven, V.V., & Wong, S.K.M. (1986). A critical analysis of vector space model for information retrieval. *JASIS*, 37 (5), 279-287.

Rama, D.V., & Srinivasan, P. (1993). An investigation of content representation using text grammars. *ACM Transactions on Information Systems*, 11 (1), 51-75.

Rau, L.F. (1992). Extracting company names from text. In *Seventh IEEE AI Applications Conference* (pp. 189-194).

Rau, L.F., & Jacobs, P.S. (1989). NL ∩ IR: natural language for information retrieval. *International Journal of Intelligent Systems*, 4, 319-343.

Rau, L.F., Jacobs, J.S., & Zemik, U. (1989). Information extraction and text summarization using linguistic knowledge acquisition. *IP&M*, 25 (4), 419-428.

Reichman, R. (1985). *Getting Computers to Talk Like You and Me: Discourse Context, Focus, and Semantics.* Cambridge, MA: The MIT Press.

Riloff, E. (1995). Little words can make a big difference for text classification. In E.A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th SIGIR Conference* (pp. 130-136). New York: ACM.

Riloff, E., & Lehnert, W. (1994). Information extraction as a basis for high-precision text classification. *ACM Transactions on Information Systems*, 12 (3), 296-333.

Ro, J.S. (1988). An evaluation of the applicability of ranking algorithms to improve the effectiveness of full-text retrieval. II. On the effectiveness of ranking algorithms on full-text retrieval. *JASIS*, 39 (3), 147-160.

Robertson, S. (1977). The probability ranking principle in IR. *Journal of Documentation*, 33, 294-304.

Robertson, S.E., & Sparck Jones, K. (1976). Relevance weighting of search terms. *JASIS*, 27 (3), 129-146.

Robertson, S.E., van Rijsbergen, C.J., &Porter, M.F. (1981). Probabilistic models of indexing and searching. In R.N. Oddy, S.E. Robertson, C.J. van Rijsbergen, & P.W. Williams (Eds.), *Information Retrieval Research* (pp. 35-56). London: Butterworths.

Robertson, S.E., & Walker, S. (1994). Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In W.B. Croft & C.J. van Rijsbergen (Eds.), *Proceedings of the Seventeenth SIGIR Conference* (pp. 232-241). London: Springer.

Robertson, S.E., & Walker, S. (1997). On relevance weights with little relevance information. In N.J. Belkin, A.D. Narasimhalu, & P. Willett (Eds.), *Proceedings of the 20th SIGIR Conference* (pp. 16-24). New York: ACM.

Robertson, S.E., Walker, S., Beaulieu, M.M., Gatford, M., & Payne, A. (1996). Okapi at TREC-4. In D.K. Harman (Ed.), *The Fourth Text REtrieval Conference (TREC-4)* (pp. 73-96). Gaithersburg, MD: NIST SP 500-236.

Rocchio, J.J. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), *The SMART Retrieval System: Experiments in Automatic Document Processing* (pp. 313-323). Englewood Cliffs, NJ: Prentice Hall.

Rowley, J.E. (1988). *Abstracting and Indexing* (2nd edition). London: Clive Bingley.

Rudolph, E. (1988). Connective relations - Connective expressions - Connective structures. In J. S. Petöfi (Ed.), *Text and Discourse Constitution: Empirical Aspects, Theoretical Approaches* (pp. 97-133). Berlin: Walter de Gruyter.

Ruge, G. (1991). Experiments on linguistically based term associations. In *RIAO 91 Conference Proceedings Intelligent Text and Image Handling* (pp. 528-545). Paris: C.I.D.-C.A.S.I.S.

Rumelhart, D.E. (1975). Notes on a schema for stories. In D.G. Bobrow & A. Collins (Eds.), *Representation and Understanding: Studies in Cognitive Science* (pp. 211-236). New York: Academic Press.

Rumelhart, D.E. (1977). *Introduction to Human Information Processing*. New York: John Wiley & Sons.

Rush, J.E., Salvador, R., & Zamora, A. (1971). Automatic abstracting and indexing. II. Production of indicative abstracts by the application of contextual inference and syntactic coherence criteria. *JASIS*, 22 (4), 260-274.

Sager, N. (1975). Sublanguage grammars in science information processing. *JASIS*, 26 (1), 10-16.

Sahami, M., Hearst, M., & Saund, E. (1996). Applying the multiple cause mixture model to text categorization. In *Machine Learning. Proceedings of the Thirteenth International Conference (ICML '96)* (pp. 435-443). San Francisco, CA: Morgan Kaufmann.

Salton, G. (1968). *Automatic Information Organization and Retrieval*. New York: McGraw-Hill Book Company.

Salton, G. (1970). Automatic text analysis. *Science*, 168, April 1970, 335-343.

Salton, G. (1971). *The SMART Retrieval System: Experiments in Automatic Document Processing*. Englewood Cliffs, NJ: Prentice Hall.

Salton, G. (1975a). A *Theory of Indexing*. Bristol, UK: J.W. Arrowsmith.

Salton, G. (1975b). Dynamic Information and Libray Processing. Englewood Cliffs, NJ: Prentice Hall.

Salton, G. (1980). Automatic term class construction using relevance - a summary of word in automatic pseudoclassification. *IP&M*, 16 (1), 1-15.

Salton, G. (1986). Another look at automatic text-retrieval systems. *Communications of the ACM*, 29 (7), 648-656.

Salton, G. (1989). *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Reading, MA: Addison-Wesley.

Salton, G., Allan, J., & Buckley, C. (1993). Approaches to passage retrieval in full text information systems. In R. Korfhage, E. Rasmussen, & P. Willett (Eds.), *Proceedings of the Sixteenth SIGIR Conference* (pp. 49-58). New York : ACM.

Salton, G., Allan, J., Buckley, C., & Singhal, A. (1994). Automatic analysis, theme generation, and summarization of machine-readable texts. *Science,* 264, 1421-1426.

Salton, G., Allan, J., Buckley, C., & Singhal, A. (1996). Automatic analysis, theme generation, and summarization of machine-readable texts. In M. Agosti & A.F. Smeaton (Eds.), *Infomation Retrieval and Hypertext* (pp. 51-96). Boston: Kluwer Academic Publishers.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *IP&M,* 24 (5), 513-523.

Salton, G., & Buckley C. (1990). Improving retrieval performance by relevance feedback. *JASIS,* 41 (4), 288-297.

Salton, G., & Buckley, C. (1991). Automatic text structuring and retrieval - experiments in automatic encyclopaedia searching. In A. Bookstein, Y. Chiaramella, G. Salton, & V.V. Raghaven (Eds.), *Proceedings of the Fourteenth SIGIR Conference* (pp. 21-30). New York: ACM.

Salton, G., Buckley, C., & Smith, M. (1990). On the application of syntactic methodologies in automatic text analysis. *IP&M,* 26 (1), 73-92.

Salton, G., & Lesk, M.E. (1971). Information analysis and dictionary construction. In G. Salton (Ed.), *The SMART Retrieval System: Experiments in Automatic Document Processing* (pp. 115-142). Englewood Cliffs, NJ: Prentice Hall.

Salton, G., & McGill, M.J. (1983). *Introduction to Modem Information Retrieval.* New York: McGraw-Hill.

Salton, G., Singhal, A., Mitra, M. & Buckley, C. (1997). Automatic text structuring and summarization. *IP&M,* 33 (2), 193-207.

Salton, G., & Yang, C.S. (1973). On the specification of term values in automatic indexing. *Journal of Documentation,* 29, 35 1-372.

Salton, G., Yang, C.S., & Yu, C.T. (1975). A theory of term importance in automatic text analysis. *JASIS,* 26 (1), 33-44.

Saracevic, T. (1975). Relevance: a review of and a framework for the thinking on the notion in information science. *JASIS,* 26, 321-343.

Saracevic, T. (1995). Evaluation of evaluation in information retrieval. In E.A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th SIGIR Conference* (pp. 138-146). New York: ACM.

Schamber, L. (1996). What is a document? Rethinking the concept in uneasy times. *JASIS,* 47 (9), 669-671.

Schamber, L., Eisenberg, M.B., & Nilan, M.S. (1990). A re-examination of relevance: toward a dynamic, situational definition. *IP&M,* 26 (6) 755-776.

Schank, R.C. (1975). *Conceptual Information Processing.* Amsterdam: North Holland.

Schank, R.C. (1982). Representing meaning: an artificial intelligence perspective. In S. Allen (Ed.), *Text Processing: Text Analysis and Generation: Text Typology and Attributes (Proceedings of NOBEL SYMPOSIUM 51)* (pp. 25-63). Stockholm, Sweden: Almqvist & Wiksell International.

Schank, R.C, & Abelson, R. (1977). *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures.* Hillsdale, NJ: Erlbaum.

Scholz, O.R. (1988). Some issues in the theory of metaphor. In J.S. Petöfi (Ed.), *Text and Discourse Constitution: Empirical Aspects, Theoretical Approaches* (pp. 269-282). Berlin: Walter de Gruyter.

Schutz, A. (1970). *Reflections on the Problem of Relevance.* New Haven, CT: Yale University Press.

Schütze, H., Hull, D.A., & Pedersen, J.O. (1995). A comparison of classifiers and document representations for the routing problem. In E.A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th SIGIR Conference* (pp. 229-237). New York: ACM.

Schütze, H., & Pedersen, J.O. (1994). A cooccurrence-based thesaurus and two applications to information retrieval. In *RIAO 94 Conference Proceedings Intelligent Multimedia Information Retrieval Systems and Management* (pp. 266-274). Paris: C.I.D.-C.A.S.I.S.

Schütze, H., Pedersen, J.O., & Hearst, M.A. (1995). Xerox TREC 3 report: combining exact and fuzzy predictors. In D.K. Harman (Ed.), *Overview of the Third Text REtrieval Conference (TREC-3)* (pp. 21-27). Gaithersburg, MD: NIST SP 500-225.

Schwarz, C. (1990). Automatic syntactic analysis of free text. *JASIS*, 41 (6), 408-417.

Scinto, L.F.M. (1983). Functional connectivity and the communicative structure of text. In J.S. Petöfi & E. Sözer (Eds.), *Micro and Macro Connexity of Text* (pp. 73-115). Hamburg: Helmut Buske.

Shannon, C.E., & Weaver, W. (1949). *The Mathematical Theory of Communication* (Eighth Printing 1959). Urbana, IL: University of Illinois Press.

Shiro, M. (1994). Inferences in discourse comprehension. In M. Coulthard (Ed.), *Advances in Written Text Analysis* (pp. 167-178). London: Routledge.

Shoham, Y. (1997). An overview of agent-oriented programming. In J.M. Bradshaw (Ed.), *Software Agents* (pp. 271- 290). Menlo Park, CA: AAAI Press.

Sidner, C.L. (1983). Focusing in the comprehension of definite anaphora. In M. Brady & R.C. Berwick (Eds.), *Computational Models of Discourse* (pp. 267-330). Cambridge, MA: The MlT Press.

Sievert, M.C. (1996). Full-text information retrieval: introduction. *JASIS*, 47 (4), 261-262.

Singhal, A., Buckley, C., & Mitra, M. (1996). Pivoted document normalization. In H.-P. Frei, D. Harman, P. Schäuble, & R. Wilkinson (Eds.), *Proceedings of the 19th SIGIR Conference* (pp. 21-29). New York: ACM.

Singhal, A., Mitra, M., & Buckley, C. (1997). Learning routing queries in a query zone. In N.J. Belkin, A.D. Narasimhalu, & P. Willett (Eds.), *Proceedings of the 20th SIGIR Conference* (pp. 25-32). New York: ACM.

Singhal, A., Salton, G., Mitra, M., & Buckley, C. (1996). Document length normalization. *IP&M*, 32 (5), 619-633.

Smeaton, A.F. (1986). Incorporating syntactic information into a document retrieval strategy: an investigation. In *Proceedings of the 1986 ACM Conference on Research and Development in Information Retrieval* (pp. 103-113). Baltimore, MD: ACM.

Smeaton, A.F. (1992). Progress in the application of natural language processing to information retrieval tasks. *Computer Journal*, 35 (3), 268-278.

Smeaton, A.F. (1996). An overview of information retrieval. In M. Agosti & A.F. Smeaton (Eds.), *Information Retrieval and Hypertext* (pp. 3-25). Boston: Kluwer Academic Publishers.

Smeaton, A.F., O'Donnell, R., & Kelledy, F. (1995). Indexing structures derived from syntax in TREC-3: system description. In D.K. Harman (Ed.), *Overview of the Third Text REtrieval Conference (TREC-3)* (pp. 55-67). Gaithersburg, MD: NIST SP 500 225.

Smeaton, A.F., & Sheridan, P. (1991). Using morpho-syntactic language analysis in phrase matching. In *RIAO 91 Conference Proceedings Intelligent Text and Image Handling* (pp. 414-429). Paris: C.I.D.-C.A.S.I.S.

Soergel, D. (1994). Indexing and retrieval performance: the logical evidence. *JASIS*, 45 (8), 589-599.

Sparck Jones, K. (1970). Some thoughts on classification for retrieval. *Journal of Documentation*, 26, 89-101.

Sparck Jones, K. (1971). *Automatic Keyword Classification for Information Retrieval.* Butterworths: London.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28 (1), 11-21,

Sparck Jones, K. (1973). Index term weighting. *Information Storage and Retrieval*, 9, 619-633.

Sparck Jones, K. (1979). Experiments in relevance weighting of search terms. *IP&M*, 15, 133-144.

Sparck Jones, K. (1991). The role of artificial intelligence in information retrieval. *JASIS*, 42 (8), 558-565.

Sparck Jones, K. (1993). What might be in a summary? In G. Knorz, J. Krause, & C. Womser-Hacker (Eds.), *Information Retrieval '93: Von der Modellierung zur Anwendung* (pp. 9-26). Konstanz: Universitätsverlag.

Sparck Jones, K., & Endres-Niggemeyer, B. (1995). Introduction automatic summarizing. *IP&M*, 31 (5), 625-630.

Sparck Jones, K., & Galliers, J.R. (1996). *Evaluating Natural Language Processing: An Analysis and Review.* New York: Springer.

Sparck Jones, K., & Tait, J.I. (1984). Automatic search term variant generation. *Journal of Documentation*, 40 (1) 50-66.

Sperber, D., & Wilson, D. (1995). *Relevance: Communication and Cognition* (2nd edition). Oxford, UK: Basil Blackwell.

Srinivasan, P. (1990). On generalizing the two-Poisson model. *JASIS*, 41 (1), 61-66.

Stadnyk, I., & Kass, R. (1992). Modelling users' interests in information filters. *Communications of the ACM*, 35 (12), 49-50.

Standera O. (1987). *The Electronic Em of Publishing. An Overview of Concepts, Technologies and Methods.* New York: Elsevier.

Stanfill, C., & Waltz, D. (1986). Towards Memory-Based Reasoning. *Communications of the ACM*, 29 (12), 1213-1228.

Strzalkowski, T. (1994). Document indexing and retrieval using natural language processing. In *RIAO 94 Conference Proceedings Intelligent Text and Image Handling* (pp. 131-143). Paris: C.I.D.-C.A.S.I.S.

Strzalkowski, T. (1995). Natural language information retrieval. *IP&M*, 31 (3), 397-417.

Strzalkowski, T., Ling, F., & Perez-Carballo, J. (1 998). Natural language information retrieval TREC-6 report. In E.M. Voorhees & D.K. Harman (Eds.), *Information Technology: The Sixth Text REtrieval Conference (TREC-6)* (pp. 347-366). Gaithersburg, MD: NIST SP 500-240.

Strzalkowski, T., Guthrie, L., Karlgren, J., Leistensnider, J., Lin, F., Perez-Carballo, J., Straszheim, T., Wang J., & Wilding, J. (1997). Natural language information retrieval: TREC-5 report. In E. Voorhees & D.K. Harman (Eds.), *Information Technology: The Fifth Text REtrieval Conference (TREC-5)* (pp. 291-3 13). Gaithersburg, MD: NIST SP 500-238.

Susskind, R. (1996). *The Future of Law: Facing the Challenges of Information Technology.* Oxford: Clarendon Press.

Suzuki, Y., Fukumoto, F., & Sekiguchi, Y. (1998). Keyword extraction of radio news using term weighting with an encyclopedia and newspaper articles. In W.B. Croft, A. Moffat, C.J. van Rijsbergen, R. Wilkinson, & J. Zobel (Eds.), *Proceedings of the 21st SIGIR Conference* (pp. 373-374). New York: ACM.

Svenonius, E. (1986). Unanswered questions in the design of controlled vocabularies. *JASIS*, 37 (5), 331-340.

Swanson, D.R. (1960). Searching natural language text by computer. *Science,* 132, October 1960, 1099-1104.

Szuprowicz, B.O. (1997). *Search Engine Technologies for the World Wide Web and Intranets.* Charleston, SC: Computer Technology Research Group.

Tait, J.I. (1985). Generating summaries using a script-based language analyser. In L. Steels & J.A. Campbell (Eds.), *Progress in Artificial Intelligence* (pp. 313-317). Chichester, UK: Ellis Horwood.

Tenopir, C. (1985). Full text database retrieval performance. *Online Review,* 9 (2), 149-164.

Tenopir, C., Ro, J.S., & Harter, S. (1991). *Full Text Databases.* London: Greenwood Press.

Teufel, S., & Moens, M. (1997). Sentence extraction as a classification task. In *Proceedings of the ACL/EACL '97 Intelligent Scalable Text Summarization Workshop.*

Thompson, P., Turtle, H., Yang, B., & Flood, J. (1995). TREC-3 Ad Hoc retrieval and routing experiments using the WIN system. In D.K. Harman (Ed.), *Overview of the Third Text REtrieval Conference (TREC-3)* (pp. 211-217). Gaithersburg, MD: NIST SP 500-225.

Tokunaga, T., Iwayama, M., & Tanaka, H. (1995). Automatic thesaurus construction based on grammatical relations. In *IJCAI-95 Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 1308-1313). San Mateo, CA: Morgan Kaufmann.

Tombros, A., & Sanderson, M. (1998). Advantages of query biased summaries in information retrieval. In W.B. Croft, A. Moffat, C.J. van Rijsbergen, R. Wilkinson, & J. Zobel (Eds.), *Proceedings of the 21st SIGIR Conference* (pp. 2-10). New York: ACM.

Tomlin, R.S., Forrest, L., Pu, M.M., & Kim, M.H. (1997). Discourse semantics. In T.A. van Dijk (Ed.), *Discourse as Structure and Process* (*Discourse Studies: A Multidisciplinary Introduction* 1) (pp. 63-111). London: SAGE.

Tong, R., Winkler, A., & Gage, P. (1993). Classification trees for document routing: a report on the TREC experiment. In D.K. Harman (Ed.), *The First Text REtrieval Conference* (pp. 209-227). Gaithersburg, MD: NIST SP 500-207.

Turtle, H.R.,& Croft, W.B. (1992). A comparison of text retrieval models. *The Computer Journal,* 35 (3), 279-290.

Tzoukermann, E., Klavans, J.L., & Jacquemin, C. (1997). Effective use of Natural Language Processing techniques for automatic conflation of multi-word terms: the role of derivational morphology, part of speech tagging, and shallow parsing. In N.J. Belkin, A.D. Narasimhalu & P. Willett (Eds.), *Proceedings of the 20th SIGIR Conference* (pp. 148-155). New York: ACM.

Uyttendaele, C., Moens, M.-F., & Dumortier, J. (1996). SALOMON: abstracting of legal cases for effective access to court decisions. In R.W. van Kralingen, H.J. van den Herik, J.E.J. Prins, M. Sergot, J. Zeleznikow (Eds.), *Proceedings of JURIX '96 Ninth International Conference on Legal Knowledge-based Systems* (pp. 47-58). Tilburg, The Netherlands: Tilburg University Press.

Uyttendaele, C., Moens, M.-F., & Dumortier, J. (1998). SALOMON: abstracting of legal cases for effective access to court decisions. *Artificial Intelligence and Law*, 6, 59-79.

Van Dijk, T.A. (1978). *Tekstwetenschap: een interdisciplinaire inleiding.* Utrecht, The Netherlands: Het Spectrum.

Van Dijk, T.A. (1985). Structures of news in the press. In T.A. van Dijk (Ed.), *Discourse and Communication: New Approaches to the Analysis of Mass Media Discourse and Communication* (pp. 69-93). Berlin: De Gruyter.

Van Dijk, T.A. (1988a). *News Analysis: Case Studies of International and National News in the Press.* Hillsdale, NJ Lawrence Erlbaum.

Van Dijk, T.A. (1988b). *News as Discourse.* Hillsdale, NJ: Lawrence Erlbaum.

Van Dijk, T.A. (1995). On macrostructures, mental Models, and other inventions: a brief personal history of the Kintsch-van Dijk Theory. In Ch. A. Weaver (III), S. Mannes, & C.R. Fletcher (Eds.), *Discourse Comprehension. Essays in Honor of Walter Kintsch* (pp. 383-410). Hillsdale, NJ: Lawrence Erlbaum.

Van Dijk, T.A. (1997). The study of discourse. In T.A. van Dijk (Ed.), *Discourse as Structure and Process (Discourse Studies: A Multidisciplinary Introduction* 1) (pp. 1-34). London: SAGE.

Van Dijk, T.A., & Kintsch, W. (1983). *Strategies of Discourse Comprehension.* New York: Academic Press.

Van Rijsbergen, C.J. (1977). A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation,* 33 (2), 106-119.

Van Rijsbergen, C. J. (1979). *Information Retrieval* (2nd ed.). London: Butterworths.

Van Rijsbergen, C.J. (1986). A non-classical logic for information retrieval. *The Computer Journal,* 29, 111-134.

Van Rijsbergen, C.J. (1989). Towards an information logic. In N.J. Belkin & C.J. van Rijsbergen (Eds.), *Proceedings of the Twelfth SIGIR Conference* (pp. 77-86). New York: ACM.

Van Rijsbergen, C.J., Harper, D.J., & Porter, M.F. (1981). The selection of good search terms. *IP&M,* 17 (2), 77-91.

Vervenne, D., Hamerlinck, F., & Vandamme, F. (1995). Electronic document management systems: an overview and future perspectives. In *Databases voor het opslaan, behandelen en beheren van documenten. De eerste stap naar een multimedia database.* Antwerpen, Belgium: Technologisch Instituut KVIV.

Voorhees, E.M. (1986). Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. *IP&M,* 22 (6), 465-476.

Voorhees, E.M. (1994). Using WordNet™ to disambiguate word senses for text retrieval. In R. Korfhage, E. Rasmussen, & P. Willett (Eds.), *Proceedings of the Sixteenth SIGIR Conference* (pp. 171-180). New York: ACM.

Voorhees, E.M., & Harman, D.K. (Eds.) (1997). *Information Technology: The Fifth Text REtrieval Conference (TREC-5).* Gaithersburg, MD: NIST SP 500-238.

Voorhees, E.M., & Harman, D.K. (Eds.) (1998). *Information Technology: The Sixth Text REtrieval Conference (TREC-6).* Gaithersburg, MD: NIST SP 500-240.

Voorhees, E.M., & Harman, D.K. (Eds.) (1999). *Information Technology: The Seventh Text REtrieval Conference (TREC-7).* Gaithersburg, MD: NIST SP 500-242.

Wade, S.J., Willett, P., & Bawden, D. (1989). SIBRIS: the sandwich interactive browsing and ranking information system. *Journal of Information Science,* 15, 249-260.

Walker, J.H. (1989). Authoring tools for complex document sets. In E. Barrett (Ed.), *The Society of Text: Hypertext, Hypermedia and the Social Construction of Information* (pp. 132-147). Cambridge, MA: The MIT Press.

Wang, Y.-C., & Vandendorpe, J. (1985). Relational thesauri in information retrieval. *JASIS,* 36 (1), 15-27.

Wang, Z.W, Wong, S.K.M., & Yao, Y.Y. (1992). An analysis of vector space models based on computational geometry. In N.J. Belkin, P. Ingwersen, & A.M. Pejtersen (Eds.), *Proceedings of the Fifteenth SIGIR Conference* (pp. 152-160). New York: ACM.

Weiss, S.M., & Indurkhya, N. (1993). Optimized rule induction. *IEEE Expert,* 8 (6), 61-69.

Weiss, S.M., & Kulikowski, C.A. (1991). *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems.* San Francisco, CA: Morgan Kaufmann.

Wellman, M.P., Durfee, E.H., & Birmingham, W.P. (1996). The digital library as a community of information agents. *IEEE Expert*, June 1996, 10-11.

Wendlandt, E.B., & Driscoll, J.R. (1991). Incorporating semantic analysis into a document retrieval strategy. In A. Bookstein, Y. Chiaramella, G. Salton, & V.V. Raghaven (Eds.), *Proceedings of the Fourteenth SIGIR Conference* (pp. 270-279). New York: ACM.

Wilbur, W.J., & Sirotkin, K. (1992). The automatic identification of stop words. *Journal of Information Science*, 18 (1), 45-55.

Wilkinson, R. (1994). Effective retrieval of structured documents. In W.B. Croft & C.J. van Rijsbergen (Eds.), *Proceedings of the Seventeenth Annual SIGIR Conference* (pp. 311-317). London: Springer.

Willett, P. (1980). Document clustering using an inverted file approach. *Journal of Information Science*, 2, 223-231.

Willett, P. (1988). Recent trends in hierarchic document clustering: a critical review. *IP&M*, 24 (5), 577-597.

Wright, P., & Lickorish, A. (1989). The influence of discourse structure on display and navigation in hypertexts. In N. Williams & P. Holt (Eds.), *Computers and Writing: Models and Tools* (pp. 90-124). Oxford, UK: Blackwell Scientific Publications.

Xu, J., & Crotl, W.B. (1998). Corpus-based stemming using cooccurrence of word variants. *ACM Transactions on Information Systems*, 16 (1), 61-81.

Yang, Y. (1994). Expert Network: effective and efficient learning from human decisions in text categorization and retrieval. In W.B. Croft & C.J. van Rijsbergen (Eds.), *Proceedings of the Seventeenth SIGIR Conference* (pp. 13-22). London: Springer.

Yang, Y. (1995). Noise reduction in a statistical approach to text categorization. In E.A. Fox, P. Ingwersen, & R. Fidel (Eds.), In *Proceedings of the 18th SIGIR Conference* (pp. 256-263). New York: ACM.

Yang, Y., & Chute, C.G. (1993). An application of least squares fit mapping to text information retrieval. In R. Korfhage, E. Rasmussen, & P. Willett (Eds.), *Proceedings of the Sixteenth SIGIR Conference* (pp. 281-290). New York: ACM.

Yang, Y., & Chute, C.G. (1994). An example-based mapping method for text categorization and retrieval. *ACM Transactions on Information Systems*, 12 (3), 252-277.

Yang, Y., Chute, C.G., Atkin, G.E., & Anda, A. (1995). TREC-3 retrieval evaluation using expert network. In D.K. Harman (Ed.), *Overview of the Third Text REtrieval Conference (TREC-3)* (pp. 299-304). Gaithersburg, MD: NIST SP 500-225.

Yang, Y., & Wilbur, J. (1996). Using corpus statistics to remove redundant words in text categorization. *JASIS*, 47 (5), 357-369.

Yochum, J.A. (1995). Research in automatic profile creation and relevance ranking with LMDS. In D.K. Harman (Ed.), *Overview of the Third Text REtrieval Conference (TREC-3)* (pp. 289-297). Gaithersburg, MD: NIST SP 500-225.

Young, S.R., & Hayes, P.J. (1985). Automatic classification and summarization of banking telexes. In *The Second Conference on Artificial Intelligence Applications: The Engineering of Knowledge Based Systems* (pp. 402-408). Washington, DC: IEEE Computer Society Press.

Zeleznikow, J., & Hunter, D. (1994). *Building Intelligent Legal Information Systems.* Deventer: Kluwer.

Zipf, G.K. (1949). *Human Behavior and the Principle of Least Effort.* Cambridge, MA: Addison-Wesley.

Zizi, M. (1996). Interactive dynamic maps for visualisation and retrieval from hypertext systems. In M. Agosti & A.F. Smeaton (Eds.), *Information Retrieval and Hypertext* (pp. 203-224). Boston: Kluwer Academic Publishers.

SUBJECT INDEX