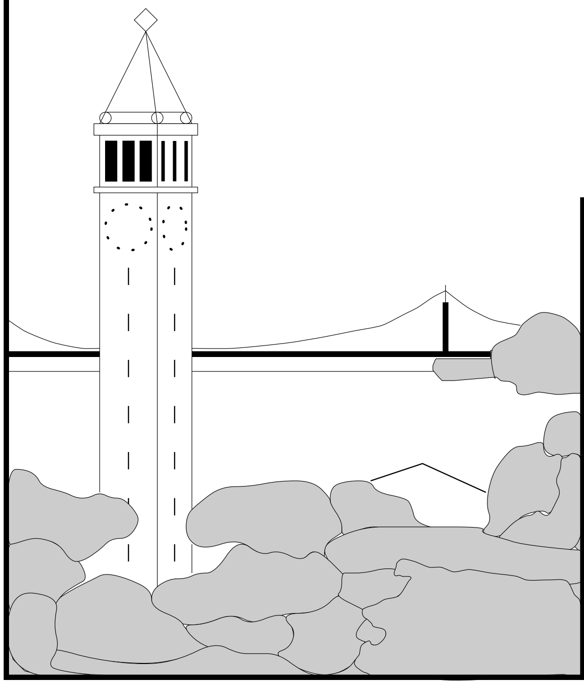
Context and Structure in Automated Full-Text Information Access

Marti A. Hearst



Report No. UCB/CSD-94/836

April 29, 1994

Computer Science Division (EECS)
University of California
Berkeley, California 94720

Context and Structure in Automated Full-Text Information Access

by

Marti A. Hearst

B.A. (University of California at Berkeley) 1985
M.S. (University of California at Berkeley) 1989

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION

of the

UNIVERSITY of CALIFORNIA at BERKELEY

Committee in charge:

Professor Robert Wilensky, Chair
Professor Ray Larson
Professor Jerome Feldman

1994

In memory of my grandfather, Alan Joseph.

Abstract

Context and Structure in Automated Full-Text Information Access

by

Marti A. Hearst

Doctor of Philosophy in Computer Science

University of California at Berkeley

Robert Wilensky
Thesis Chair

This dissertation investigates the role of contextual information in the automated retrieval and display of full-text documents, using robust natural language processing algorithms to automatically detect structure in and assign topic labels to texts. Many long texts are comprised of complex topic and subtopic structure, a fact ignored by existing information access methods. I present two algorithms which detect such structure, and two visual display paradigms which use the results of these algorithms to show the interactions of multiple main topics, multiple subtopics, and the relations between main topics and subtopics.

The first algorithm, called *TextTiling*, recognizes the subtopic structure of texts as dictated by their content. It uses domain-independent lexical frequency and distribution information to partition texts into multi-paragraph passages. The results are found to correspond well to reader judgments of major subtopic boundaries. The second algorithm assigns multiple main topic labels to each text, where the labels are chosen from pre-defined, intuitive category sets; the algorithm is trained on unlabeled text.

A new iconic representation, called *TileBars* uses TextTiles to simultaneously and compactly display query term frequency, query term distribution and relative document length. This representation provides an informative alternative to ranking long texts according to

their overall similarity to a query. For example, a user can choose to view those documents that have an extended discussion of one set of terms and a brief but overlapping discussion of a second set of terms. This representation also allows for relevance feedback on patterns of term distribution.

TileBars display documents only in terms of words supplied in the user query. For a given retrieved text, if the query words do not correspond to its main topics, the user cannot discern in what context the query terms were used. For example, a query on *contaminants* may retrieve documents whose main topics relate to nuclear power, food, or oil spills. To address this issue, I describe a graphical interface, called *Cougar*, that displays retrieved documents in terms of interactions among their automatically-assigned main topics, thus allowing users to familiarize themselves with the topics and terminology of a text collection.

Contents

Acknowledgments

Research is a surprisingly social activity. Graduate school has been an enormously positive experience, in large part because of the outstanding people around me.

I would like to thank my advisor, Robert Wilensky, for his supportive, unflagging enthusiasm for this work, his helpful analytic insights, and his willingness to let me choose some of my own paths. His comments have greatly improved this thesis, and he was extraordinarily fast at reading and returning chapters to me, despite his pressing duties as Department Chair.

I would also like to thank the other members of my committee: Ray Larson, for contributing his expertise in information retrieval, and Jerry Feldman. Dan Jurafsky read and critiqued this entire thesis, and for this he has my profuse thanks. Narciso Jaramillo also provided many helpful comments at the final hour.

Mike Stonebraker encouraged me to enter graduate school in computer science, wrote the crucial letter, convinced me to come to Berkeley, gave me a research assistantship under which I earned my master's degree, and even supported me for a time after I switched fields. In many ways Mike is a visionary and his attitudes about the field and how to do research have strongly influenced me.

Peter Norvig, who knows about all NLP work ever done (and has implemented a large fraction if it) has helped me out at several critical strategic junctions, and tried hard not to look at me funny the first time I mentioned "big text".

I cannot begin to state the importance of my continued association with the Xerox Palo Alto Research Center. I owe an enormous debt to Per-Kristian Halvorsen, a Montague-semantician who liked my off-the-wall ideas about cognitive linguistics and invited me to spend my first summer at PARC. Jan Pedersen, as project leader for the information access group, and as a friend, has been constantly supportive, and has answered innumerable questions about statistics. Over the past five years, the Thursday Reading Group has provided a thought-provoking but lighthearted forum for discussion of computational linguistics and information access. In addition to Jan and Per-Kristian, this insightful group has included Francine Chen, Doug Cutting, Mary Dalrymple, Ken Feuerman, David Hull, Ron Kaplan, Laurie Karttunen, Martin Kay, David Karger, Julian Kupiec, Chris Manning, John Maxwell, Hinrich Schütze, Penni Sibun, John Tukey, Lynn Wilcox, Meg Withgott, and Annie Zaenen. Jeanette Figueroa is the most effective, efficient, and affectionate administrative assistant that anyone could ever hope to work with. Mark Weiser is a model to emulate.

Daily life in graduate school sparkled in The Office of All Topics (no topic is taboo). I will miss the analysis sessions and the laughter shared with Michael Braverman, nj Jaramillo, and Mike Schiff, not to mention their skills at paper critiquing, talk debugging, and question answering.

My other Berkeley colleagues in (computational) linguistics – Jane Edwards, Adele Goldberg, Dan Jurafsky, Steve Omohundru, Terry Regier, Andreas Stolcke, and Dekai Wu – are simultaneously brilliant and fun and have greatly enriched my understanding of the field.

I am the last of the four "mars" to graduate: Mary Gray Baker, Marie desJardins and Margo Seltzer have been invaluable friends and colleagues. Others on campus that have provided support, advice, and friendship include Nina Amenta, Elan Amir, Paul Aoki, Francesca Barrientos, Marshall Bern, Eric Enderton, Mark Ferrari, John Hartman, Mor Harchol, Chris Hoadley, Kim Keeton, Adam Landsberg, Steve Lucco, Bruce Mah, Nikki Mirghafori, Sylvia Plevritis, Patti Schank, Mark Sullivan, Seth Teller, Tara Weber, and Randi Weinstein.

I have been privileged to learn about language and the mind from Chuck Fillmore, Alison Gopnik, Paul Kay, George Lakoff, John Searle, and Dan Slobin. Michael Ranney has been especially supportive and informative about Cognitive Science. Ray Larson and Bill Cooper of the Berkeley School for Library Studies have both generously shared their advice, knowledge, and books on information access, and Michael Buckland and Cliff Lynch have kept me informed of the cutting edge of the field. In the Computer Science department, Tom Anderson, Mike Clancy, Jim Demmel, Randy Katz, John Ousterhout and Kathy Yelick have been helpful and approachable, and I greatly admire Dave Patterson's confidence and optimism as both a leader and a researcher.

Ken Church helped pioneer the field of corpus-based computational linguistics, and I owe him a special thanks for inviting me to an instructive summertime visit at AT&T Bell Laboratories. I would also like to thank Paul Jacobs for organizing the 1990 AAAI Spring Symposium on Text-based Intelligent Systems, which was a watershed event in the course of my research.

I have benefitted from interactions with my coauthors Anne Fontaine, Greg Grefenstette, David Palmer, Chris Plaunt, Philip Resnik, and Hinrich Schütze, and conference-colleagues Ido Dagan, Haym Hirsh and David Yarowsky. I am also grateful for the advice and opinions of Bill Gale, Graeme Hirst, and Gerry Salton.

John Ousterhout has done everyone a great service by inventing Tcl/Tk. I'm also grateful to Ethan Munson for maintaining and customizing the latex thesis style files, and Dan Jurafsky for creating the lsalike bibtex style.

Sheila Humphreys has been infallibly supportive, including finding financial support for me during a tricky period. Kathryn Crabtree makes splendid the difference between attending Berkeley as an undergraduate and as a graduate computer science major. Liza Gabato, Jean Root, Teddy Diaz, and Crystal Williams of the CS Department staff have been very helpful through the years.

Friends from the outside world who have stuck with me through this include Irene Fong, Jane Choi Greenthal, Annie and Bret Peterson, Kayako Shimada, Terry Tao, Greg Theisen and Susan Wood.

John Batali taught me about philosophy, feminism, and dinosaurs.

This research has been supported at various times by the following sources of funding
(in order of appearance):
The U.S. Air Force Office of Scientific Research, Grant 83-0254 and the Naval Electronics Systems Command Contract N39-82-C-0235.
The Army Research Organization, Grant DAAL03-87-0083 and the Defense Advanced Research Projects Agency through NASA Grant NAG 2-530.
A California Legislative Grant.
Digital Equipment Corporation under Digital's flagship research project Sequoia 2000: Large Capacity Object Servers to Support Global Change Research.
AT&T Bell Laboratories.
The Advanced Research Projects Agency (ARPA) under Grant No. MDA972-92-J-1029 with the Corporation for National Research Initiatives (CNRI).

The Xerox Palo Alto Research Center has supported my work from 1989 to the present.

I'd like to thank my brother Ed ("roll with the punches") and my sister Dor for being sympathetic, and Grandma Mar for encouraging us to do what makes us happy. Finally, I'd like to thank my parents for their undivided love and support and all those Sunday night dinners in Berkeley.

Chapter 1

Introduction

1.1 Full-Text Information Access

Full-length documents have only recently become available online in large quantities, although bibliographic records and technical abstracts have been accessible for many years (Tenopir & Ro 1990). For this reason, information retrieval research has mainly focused on retrieval from titles and abstracts. In this dissertation, I argue that the advent of full-length text should be met with new approaches to text analysis, particularly for the purposes of information access.[1] I emphasize that, for the purposes of information access, full text requires context, that is, the mechanisms used for retrieval and display of full-text documents should take into account the context in which the query terms and document terms are used. Each chapter of this thesis discusses some aspect of supplying or using contextual information in order to facilitate information access from full text documents.

This emphasis on context in full-text information access arises from the observation that full text is qualitatively different from abstracts and short texts. Most of the content words in an abstract are salient for retrieval purposes because they act as placeholders for multiple occurrences of those terms in the original text, and because these terms tend to pertain to the most important topics in the text. On the other hand, in a full-text document, many terms occur which do not represent the essence of the main contents of the text. Expository texts such as science magazine articles and environmental impact reports can be viewed as consisting of a series of short, sometimes densely discussed, subtopics that are understood within the context of the main topics of the texts.

Consider a 23-paragraph article from *Discover* magazine. A reader divided this text into the segments of Figure 1.1, with the labels shown, where the numbers indicate paragraph numbers. The main topic of this text is the exploration of Venus by the space probe Magellan. There are also several subtopical discussions that cover more than one paragraph. These

[1] The term *information access* is beginning to supercede that of *information retrieval* since the latter's implication is too narrow; the field should be concerned with information retrieval, display, filtering, and query facilitation.

1- 2	*Intro to Magellan space probe*
3	*Atmosphere obscures view*
4	*Climate*
5- 7	*Meteors*
8-11	*Volcanic activity*
12-15	*Styx channel*
16-17	*Aphrodite Highland*
18	*Gravity readings*
19-21	*Recent volcanic activity*
22-23	*Future of Magellan*

Figure 1.1: Paragraph-level breakdown of the subtopic structure of an expository text.

include a discussion of evidence for volcanic activity on Venus and a discussion of a large channel known as the River Styx. If the topic "volcanic activity", or perhaps "geological activity", is of interest to a user, an information access system must decide whether or not to retrieve this document. Since volcanism is not a main topic, the frequencies of use of this term will not dominate the statistics characterizing the document; therefore, to find "volcanic activity" in this case, a system will have to retrieve documents in which the terms of interest are not the most frequent terms in the document. On the other hand, the system should not necessarily select a document just because there are a few references to the target terms. Information about the topic structure would allow a distinction to be made between main topics, subtopics, and passing references. Thus there is a need for identifying the topic structure of documents.

In this dissertation I suggest that the relative distribution of terms within a text provides clues about its main topic and subtopic structure, and that this information should be made explicit and available to the users of a full-text information access system.

One way to try to determine if two terms occur in the same subtopic or in some other co-modificational relationship is to observe whether both occur in the same passage of the text. However, the notion of "passage" is not well defined. (In many cases author-defined sectioning information is not present or is too coarse-grained.) A simple assumption is that every paragraph is a passage and every passage is a paragraph. But often the contents of a long text can be understood in terms of groupings of adjacent paragraphs, as seen in the example above. This observation opens a new question for computational linguistics: how can multiple-paragraph passages be automatically identified?

A simple approach is to divide documents into approximately even-sized, but arbitrarily chosen, multi-paragraph pieces. A more appealing, but less straightforwardly automatizable approach is to group paragraphs together that discuss the same subtheme or subtopic. This dissertation describes a fully-implemented, domain-independent text analysis approach

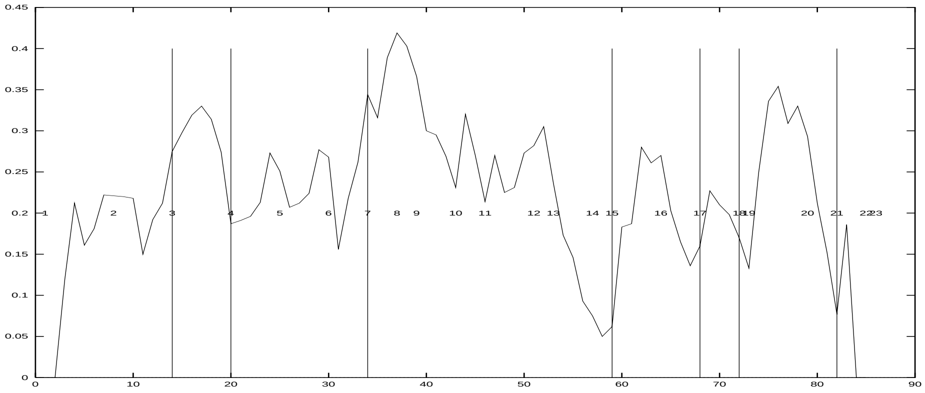


Figure 1.2: The output of the TextTiling algorithm when run on the Magellan Text. Internal numbers indicate paragraph numbers. Vertical lines indicate boundaries chosen by the algorithm; for example, the leftmost vertical line represents a boundary after paragraph 3. Note how these align with the outline of the Magellan text in Figure 1.1.

called *TextTiling* that attempts this task. The TextTiling algorithm makes use of lexical cohesion relations to recognize where subtopic changes occur. For a given block size, the algorithm compares the lexical similarity of every pair of adjacent blocks. The resulting similarity scores are plotted against sentence number, and after being graphed and smoothed, the plot is examined for peaks and valleys (see Figure 1.2). High similarity values, implying that the adjacent blocks cohere well, tend to form peaks, whereas low similarity values, indicating a potential boundary between TextTiles, create valleys. The algorithm's results fit between upper and lower evaluation bounds, where the upper bound corresponds to reader judgments and the lower bound is a simple, reasonable approach to the problem that can be automated. TextTiling is discussed in Chapter 2.

By casting document content in terms of topical structure, I have developed new ideas about the role of document structure in information access. An inherent problem with information retrieval ranking functions is they make a decision about the criteria upon which documents are ranked which is opaque to the user. This is especially problematic when performing a retrieval function other than full similarity comparison since query terms can have many different term distribution patterns within a full-text document, and different patterns may imply different semantics. In some cases a user might like to find documents that discuss one term as a main topic with perhaps just a short discussion of another term as a subtopic. Current information access paradigms provide no way to express this kind of preference. To help remedy this, I present a new representational paradigm, called *TileBars*, which provides a compact and informative iconic representation of the documents' contents with respect to the query terms (see Figure 1.3). TileBars allow users to make informed decisions about not only which documents to view, but also which passages of those documents to select, based on the distribution of the query terms in the documents.

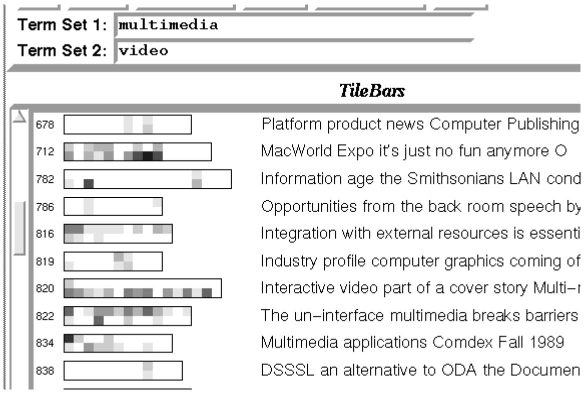


Figure 1.3: TileBars for a query in which the terms *multimedia* and *video* are contrasted. Rectangles correspond to documents, squares correspond to TextTiles, the darkness of a square indicates the frequency of terms in the corresponding Term Set. The title and initial words of a document appear next to its TileBar.

TileBars use TextTiles to break documents into coherent subparts. The query term distribution is computed for each document and the resulting frequency is indicated for each tile, in a bar-like image. The bars for each set of query terms are displayed in a stacked sequence, yielding a representation that simultaneously and compactly indicates relative document length, query term frequency, and query term distribution. The representation exploits the natural pattern-recognition capabilities of the human perceptual system; the patterns in a column of TileBars can be quickly scanned and deciphered.

TileBars support a paradigm in which the system does not decide on a single ranking strategy in advance, but instead provides information that allows the user to determine what kind of distributional relationships are useful. Chapter 3 describes TileBars and their uses, as well as other issues relating to passage retrieval.

TileBars display documents only in terms of words supplied in the user query. For a given retrieved text, if the query words do not correspond to its main topics, the user cannot discern the context in which the query terms were used. For example, a query on *contaminants* may retrieve documents whose main topics relate to nuclear power, food, or oil spills. To help account for this, I suggest assigning to each text category labels that correspond to its main topics, so that users can get a feeling for the domain in which query terms are to be used. Thus if two documents discuss the same main topic themes but use different terms to do so, one unified category can be used to represent their content. Similarly, if a document uses many different terms to build up the impression of a theme, then the category can capture this information in a compact form. If a document is best described by more than one category, it can be assigned multiple categories, and two documents that share one major theme but do not share others can be shown to be related only along the one shared dimension.

Toward this end, Chapter 4 describes an algorithm that automatically assigns main topic

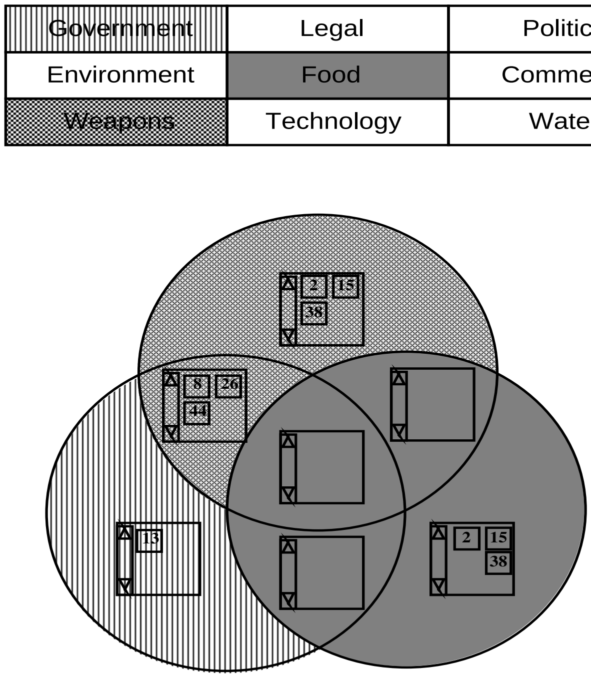


Figure 1.4: A sketch of the Cougar interface; three topic labels have been selected.

category labels to texts, and Chapter 5 presents a new display mechanism for making this information available to the user. The categorization algorithm uses a large text collection to determine which terms are salient indicators for each category. The algorithm also allows for the existence of multiple simultaneous themes since each word in the text can contribute to evidence for a category model, and each word can contribute evidence to more than one model, if appropriate. One of the category sets used by the algorithm consists of 106 general-interest categories; Chapter 4 describes an algorithm that automatically derives these categories from an existing hierarchical lexicon.

Once multiple main topic categories have been assigned to a text, they must be displayed effectively. Chapter 5 describes an interface called *Cougar* in which fixed category sets are used for two purposes: to orient the user to the dataset under scrutiny, and to place the results of the query into context (see Figure 1.4). Cougar allows users to view retrieved documents in terms of the interaction among their main topics, using the categorization algorithm from Chapter 4 to provide contextual information. The interface helps users become familiar with the topics and terminology of an unfamiliar text collection. A consequence of allowing multiple topics per document is that the display must handle multi-dimensional information. The approach used here again allows user input to play a role: the user specifies which categories are at the focus of attention at any given time. Cougar supplies a simple mechanism of visual intersection to allow users to understand how the retrieved documents are related to one another with respect to their main topic categories.

1.2 An Approach to Computational Linguistics

One goal of natural language processing is to design programs which interpret texts in much the same way that a human reader would. Since this is such a difficult task and since it requires a large amount of domain knowledge, most of the work of this sort focuses on small collections of sentences. This approach is appropriate when automating detailed text interpretation (e.g.,Schank & Abelson (1977), Wilks (1975), Wilensky (1983a), Charniak (1983), Norvig (1987)) or when supporting a theory about human inference and parsing mechanisms (e.g., Martin (1990), Jurafsky (1992)), but with some exceptions the state of the art is such that the use of this kind of analysis in information access is still a distant goal.

In the past five years there has been an increasing tendency to take a data-intensive approach to language analysis, focusing on broad but coarse-grained coverage of unrestricted text (Church & Mercer 1993). This approach is still uncommon in the area of discourse analysis; the work here is an exception. The algorithms presented here are domain-independent but approximate, scalable but error-prone, in the hopes that their application to the coarser goals of information access will nevertheless be useful. Such approximate methods seem especially appropriate for text segmentation, and information access more generally. These are intrinsically "fuzzy" tasks, in the sense that they generally have no objectively correct answer, and many different results may be deemed reasonable (compared with, for example, grammaticality judgments). Readers often disagree about where to draw a boundary marking a topic shift, or whether a given text is relevant to a query; therefore it seems implausible to expect exact answers to such questions. This thesis demonstrates that despite the inherent plasticity of these tasks, automating these processes can still yield useful results.

Chapter 2

TextTiling

2.1 Introduction: Multi-paragraph Segmentation

The structure of expository texts can be characterized as a sequence of subtopical discussions that occur in the context of a few main topic discussions. For example, a popular science text called *Stargazers*, whose main topic is the existence of life on earth and other planets, can be described as consisting of the following subdiscussions (numbers indicate paragraph numbers):

1-3	*Intro – the search for life in space*
4-5	*The moon's chemical composition*
6-8	*How the early proximity of the moon shaped it*
9-12	*How the moon helped life evolve on earth*
13	*The improbability of the earth-moon system*
14-16	*Binary/trinary star systems make life unlikely*
17-18	*The low probability of non-binary/trinary systems*
19-20	*Properties of our sun that facilitate life*
21	*Summary*

Subtopic structure is sometimes marked in technical texts by headings and subheadings which divide the text into coherent segments; Brown & Yule (1983:140) state that this kind of division is one of the most basic in discourse. However, many expository texts consist of long sequences of paragraphs with very little structural demarcation.

This chapter describes why such structure is useful and presents algorithms for automatically detecting such structure.[1] Because the model of discourse structure is one in which text is partitioned into contiguous, nonoverlapping blocks, I call the general approach

[1] I am grateful to Anne Fontaine for her interest and help in the early stages of this work.

TextTiling. The ultimate goal is to not only identify the extents of the subtopical units, but to label their contents as well. This chapter will focus only on the discovery of subtopic structure, leaving determination of subtopic content to future work. (Chapter 4 discusses automatic assignment of main topic categories.)

2.2 What is Subtopic Structure?

In order to describe the detection of subtopic structure, it is important to define the phenomena of interest. The use of the term "subtopic" here is meant to signify pieces of text 'about' something (and is not to be confused with the topic/comment (Grimes 1975) distinction found within individual sentences). The intended sense is that described in Brown & Yule (1983:69):

> In order to divide up a lengthy recording of conversational data into chunks which can be investigated in detail, the analyst is often forced to depend on intuitive notions about where one part of a conversation ends and another begins. ... Which point of speaker-change, among the many, could be treated as the end of one chunk of the conversation? This type of decision is usually made by appealing to an intuitive notion of **topic**. The conversationalists stop talking about 'money' and move on to 'sex'. A chunk of conversational discourse, then, can be treated as a unit of some kind because it is on a particular 'topic'. The notion of 'topic' is clearly an intuitively satisfactory way of describing the unifying principle which makes one stretch of discourse 'about' something and the next stretch 'about' something else, for it is appealed to very frequently in the discourse analysis literature.
>
> Yet the basis for the identification of 'topic' is rarely made explicit.

Others who have stated the intended sense include Rotondo (1984), who writes "A macro-unit can be roughly defined as any coherent subpart of a text which is assigned a global interpretation of its own" and Tannen (1984:38, cited in Youmans (1991)) who, when discussing spoken discourse, claims: "... the most useful unit of study turned out to be the episode, bounded by changes of topic or activity, rather than, for example, the adjacency pair or the speech act."

Hinds (1979:137) suggests that different discourse types have different organizing principles. TextTiling is geared towards expository text; that is, text that explicitly explains or teaches, as opposed to, say, literary texts. More specifically, TextTiling is meant to apply to expository text that is not heavily stylized or structured. A typical example is a five-page science magazine article or a twenty-page environmental impact report. It excludes documents composed of short "news bites" or any other disjointed, although lengthy, text.

A two-level structure is chosen for reasons of computational feasibility and to coincide with the goals of the use of the algorithms' results. This thesis employs only algorithms that can be implemented, that can be run on real texts, and that can run on a variety of texts independent of their domain of discourse. Given the current state of the art, this can best be done by methods that work in a coarse way on coarse units of information. The applications for which the results are to be used do not necessarily require fine-grained distinctions. This is especially true of some kinds of information retrieval applications. A user might have difficulty formulating a query in which multiple embedded levels of topic structure need be specified, although this kind of information could be useful for browsing. Most existing approaches to discourse processing are too ambitious to yield generally applicable results; it is hoped that by trying to make coarser distinctions the results will be more universally successful.

2.3 Why Multi-Paragraph Units?

In school we are didactically taught to write paragraphs in a certain form; therefore a common assumption is that most paragraphs have a certain kind of well-formed structure, complete with topic sentence and summary sentence. In real-world text, these expectations are often not met. But even if a paragraph is written in a self-contained, encapsulated manner, a particular subtopical discussion can span multiple paragraphs, with only different nuances being discussed in the paragraphs that comprise the discussion.

Multi-paragraph segmentation has many potential applications, including:

- Information Access
- Corpus-based Computational Linguistics
- Text Display and Hypertext
- Text Summarization

Applications to information access are a major concern of this thesis and are discussed in detail in Chapter 3. There, I describe how tiles are used in an iconic graphical representation that allows the user to understand the distributional relationships between terms in a query and terms in the retrieved documents. Another benefit of using multi-paragraph segmentation is that since in most cases there are fewer tiles per document than paragraphs, tiles require less storage and comparison time for otherwise equivalent, paragraph-based algorithms.

However, multi-paragraph segmentation has broader applications. These are described below.

2.3.1 Corpus-based Computational Linguistics

An increasingly important algorithmic strategy in computational linguistics is to derive information about the distributional patterns of language from large text collections, or corpora. Several such algorithms make use of information about lexical co-occurrence; that is, they count how often terms occur near one another across many texts.

Some of these algorithms use only very local context. For example, working with large text collections, Brent (1991) and Manning (1993) make use of restricted syntactic information to recognize verb subcategorization frames, Smadja & McKeown (1990) create collections of collocations by gathering statistics about words that co-occur within a few words of one another, and Church & Hanks (1990) use frequency of co-occurrence of content words to create clusters of semantically similar words.

However, several algorithms gather co-occurrence statistics from large windows of text, usually of fixed length. For example, the disambiguation algorithms of Yarowsky (1992) and Gale *et al.* (1992b) train on large, fixed-sized windows of text. In these algorithms, all terms that reside within a window of text are grouped together to supply evidence about the context in which a word sense occurs. For example, an instance of the tool sense of the word *crane* might be surrounding by terms associated with large mechanical tools, such as *lift* and *construction*. Terms surrounding the bird sense would tend to be those more associated with birddom. A question arises about how much context surrounding the target word should be included in the association. Gale *et al.* (1992b) have shown that, at least in one corpus, useful sense information can extend out for thousands of words from the target term. In practice Yarowsky (1992) uses a fixed window of 100 words.

Gale *et al.* (1992c) and Gale *et al.* (1992a) provide evidence that in most cases only one sense of a word is used in a given discourse. For example, if the word *bill* is used in its legislative sense in a discourse, then it is unlikely to be used in the sense of the body part of a duck in that same discourse. They performed experiments which indicate that the same sense of a polysemous word occurred throughout encyclopedia articles and Canadian parliament proceedings. It is possible that in texts whose contents are less stereotyped, different senses of the same word will occur, but in different contexts within the same text, that is, not particularly near one another. If this is the case, then motivated multi-paragraph segmentation could help determine the boundaries within which single senses of polysemous words are used.

Another example of an algorithm that derives lexical co-occurrence information is Word Space (Schütze 1993b). In this algorithm, statistics are collected about the contexts in which words co-occur. The results are placed in a term-by-term co-occurrence matrix which is then reduced using a variant of multidimensional scaling. The resulting matrix can be used to make inferences about the closeness of words in a multidimensional semantic space. Currently the co-occurrence information is found by experimenting with different fixed window sizes and chosing one that works best for a test set.

A critical assumption underlying these algorithms is that the terms co-occurring within the text window do so because they are at least loosely semantically related. It seems plausible that changes in discourse structure will correspond to changes in word usages, and so the quality of the statistics for these algorithms should benefit from the use of training texts that have been partitioned on the basis of subtopic content.

2.3.2 Online Text Display and Hypertext

Research in hypertext and text display has produced hypotheses about how textual information should be displayed to users. One study of an online documentation system (Girill 1991) compared display of fine-grained portions of text (i.e., sentences), full texts, and intermediate sized units. Girill found that divisions at the fine-grained level were less efficient to manage and less effective in delivering useful answers than intermediate sized units of text. (Girill also found that using document boundaries is more useful than ignoring document boundaries, as is done in some hypertext systems.) The author does not make a commitment about exactly how large the desired text unit should be, instead talking about "passages" and describing passages in terms of the communicative goals they accomplish (e.g., a problem statement, an illustrative example, an enumerated list). The implication is that the proper unit is the one that groups together the information that performs some communicative function; in most cases this unit will range from one to several paragraphs. (Girill implies that pre-marked sectional information, if available and not too long, is an appropriate unit.)

Tombaugh *et al.* (1987) explore issues relating to ease of readability of long texts on CRT screens. Their study explores the usefulness of multiple windows for organizing the contents of long texts, hypothesizing that providing readers with spatial cues about the location of portions of previously read texts will aid in their recall of the information and their ability to quickly locate information that has already been read once. In the experiment, the text is divided into pre-marked sectional information, one section placed in each window. They conclude that segmenting the text by means of multiple windows can be very helpful if readers are familiar with the mechanisms supplied for manipulating the display.

Converting text to hypertext in what is called post-hoc authoring (Marchionini *et al.* 1991) requires division of the original text into meaningful units (a task noted by these authors to be a challenging one) as well as meaningful interconnection of the units. Automated multi-paragraph segmentation should help with the first step of this process.

2.3.3 Text Summarization and Generation

Nineteenth century histories and travelogues often prefaced chapters with a list of topical discussions, providing a guide for the reader as to the contents to come. These descriptions are not abstracted summaries, but rather are lists of the subdiscussions that take place during the course of the chapter. For example, Chapter 1 of Alexis de Tocqueville's *Democracy in America, Volume 1* is entitled "Exterior Form of North America" and is prefaced with the following text:

> North America divided into two vast regions, one inclining towards the Pole, the other towards the Equator – Valley of the Mississippi – Traces found there of the revolutions of the globe – Shore of the Atlantic Ocean, on which the English

01-06 *North America divided into two vast regions, one inclining towards the Pole, the other towards the Equator*
07-09 *Valley of the Mississippi*
10-11 *Traces found there of the revolutions of the globe*
12-13 *Shore of the Atlantic Ocean, on which the English colonies were founded*
14-16 *Different aspects of North and of South America at the time of their discovery*
17-18 *Forests of North America*
19-19 *Prairies*
21-25 *Wandering tribes of natives*
20-20 *Their outward appearance, customs, and languages*
26-28 *Traces of an unknown people.*

Figure 2.1: Paragraph-level breakdown of the subtopic structure of Tocqueville Chapter 1, Volume 1.

> colonies were founded – Different aspects of North and of South America at the time of their discovery – Forests of North America – Prairies – Wandering tribes of natives – Their outward appearance, customs, and languages – Traces of an unknown people.

These descriptions can be construed to be subtopical discussions that take place in the context of a discussion of the exterior form of North America. The list closely reflects the order of discussion of the subtopics in the ensuing chapter, with a few exceptions of order switchings and paragraphs whose content plays a bridging role and so does not merit mention in the subtopic list. Figure 2.1 below shows that the subtopic discussions in most cases span more than one paragraph. Although the paragraphs in and of themselves are somewhat encapsulated, this example demonstrates that the multi-paragraph unit size can indeed be a meaningful one.

A scan of the subtopic discussions makes it apparent that the title of the chapter does not adequately cover the contents of the text. A discussion of the early inhabitants of the continent is not something one tends to classify as central to its exterior form. The title might better be served as "Exterior Form and Early Inhabitants of North America". The assumption that a logical text unit must discuss only one topic might be at least partly responsible for the mistitle.

Multi-paragraph subtopic structure should act as a first step toward automatic determination of text synopses. Algorithms that extract salient phrases from texts in order to create synopses (e.g., Chen & Withgott (1992), Pollock & Zamora (1975)) currently do not usually take this kind of information into account. Paice (1990) recognizes the need for taking topical structure into account but does not have a method for determining such structure.

An interesting alternative approach appears in the work of Alterman & Bookman (1990). The authors apply knowledge-intensive techniques to interpret short texts and then plot the number of inferences that can be made against the clausal position in the text. They use the resulting plot to determine the "thickness" of the text at each point; breaks in thickness indicate an episode change. Summaries are produced by finding the main episode boundaries and extracting concepts from each episode that is deemed to be important (using another measure). Although the technique is heavily knowledge-oriented and computationally expensive, and the length of each episode is about two sentences on average, the general idea bears some resemblance to that discussed below.

Turning now to the related topic of text generation, Mooney *et al.* (1990) assert that the high level structure of extended explanations is determined by processes separate from those which organize text at lower levels. They present a scheme for text generation that is centered around the notion of Basic Blocks: multi-paragraph units of text, each of which consists of (1) an organizational focus such as a person or a location, and (2) a set of concepts related to that focus. Thus their scheme emphasizes the importance of organizing the high level structure of a text according to its topical content, and afterwards incorporating the necessary relatedness information, as reflected in discourse cues, in a finer-grained pass. This use of multi-paragraph units for coherent generation implies that this unit of segmentation should be useful in recognition tasks as well.

2.4 Discourse Structure

When analyzing textual discourse structure, two important and related issues are: what kind of structure is inherent in discourse, and what mechanisms and aspects of language are needed to detect that structure. Although the second is strongly influenced by the first, it is not unambiguously determined by the first; that is, one kind of structure can be recognized via lexical distribution patterns, isolated discourse cues, and other factors, with varying degrees of success.

Two important subissues arise with respect to the choice of assumptions about the structure of discourse:

1. At what level of granularity are the units of the discourse? Is the salient unit the word, phrase, clause, sentence, paragraph, or something else? Is more than one level of granularity appropriate?

2. What is the topology of the discourse structure? I.e., what form do the patterns of interrelations among the units of the discourse structure take?

The nature of the analysis can be heavily dependent on whether or not the theory is geared towards a computational versus an analytical framework. An additional influential factor is the perceived role or purpose of the discourse structure. If the goal of discourse analysis is to allow the system to answer questions in an interactive session with a human, then issues such as the intentions of the speakers must be taken into account (e.g., Wilensky *et al.* (1984), Moore & Pollack (1992)). Researchers working on tutoring and advice systems that engage in dialogues with humans have tended to emphasize pragmatics, e.g., reference resolution. This usually requires an understanding of issues relating to discourse focus and centering. An important aspect of Winograd's classic thesis work (Winograd 1972) is his program's ability to determine which object is the one most likely to be under discussion. He does this by incorporating a variety of factors, including the current context and focus of the discourse as well as the semantics of the objects and relationships under discussion (cf. §8.2). In spoken-text discourse analysis, focus is usually studied at the sentential level, with links among foci typically spanning only a few sentences. Other examples are the computational work of Grosz (1986) and Sidner (1983), who examine issues relating to focus and anaphor resolution.

Other research emphasizes the syntactic aspects of anaphor resolution and ellipsis, for example, Dalrymple *et al.* (1991) and Hardt (1992). Another approach is the application of plans, e.g., Wilensky (1981), Lambert & Carberry (1991) and knowledge, e.g., Hobbs (1978), Luperfoy (1992), Cardie (1992), to anaphor resolution and other interpretation tasks.

As is evident from the discussion above, a large part of the computational discourse work has been done in the context of interactive systems. In general, the discourse characteristics of spoken text are quite different from those of written, especially expository, text (Brown & Yule 1983) (§1.2). The goals of analyzing texts for interactive systems are different from those of discourse segmentation of written texts into subtopical boundaries, and it follows that the choice of discourse unit and topology differ for the different tasks.

2.4.1 Granularity of Discourse Structure

There is a tradition in linguistics of viewing discourse structure as the study of relations at the interphrasal or interclausal level. The notion of the given/new (or topic/comment) distinction is an extensively studied one in linguistics. In English, topics, in this sense, are usually subjects and comments are the associated predicates. In some languages the distinction is marked more overtly (Kuno 1972), (Grimes 1975). This is closely related to the distinctions of theme/rheme and given/new at the sentential level.

Work on prosodic structure of spoken text usually takes place at the inter-sentential level, e.g., Wang & Hirschberg (1992), Bachenko *et al.* (1986). As mentioned above, work in anaphora resolution tends to focus on intra-sentential units, as does most text-generation work.

The hierarchical theories of discourse such as the theory of attentional/intentional structure (Grosz & Sidner 1986), and Rhetorical Structure Theory (Mann & Thompson 1987) tend to use phrasal or clausal units as building blocks from which analyses of length from one to three paragraphs long are made (for example, in Morris (1988), intentional structure is found for texts of approximately 40 sentences in length).

Discourse work at the multi-paragraph level has been mainly in the theoretical realm, notably the work on macrostructures (van Dijk 1980) (van Dijk 1981) and the work on story grammars (Lakoff 1972),(Rumelhart 1975). An exception is the work of Batali (1991) that makes use of discourse structure in the automated interpretation of (simplified) chapters of introductory physics texts, with the goal of learning rules for solving problems in kinematics.

2.4.2 Topology of Discourse Structure

Hierarchical Models

Many theories of discourse structure, both computational and analytical, assume a hierarchical model of discourse. Two prominent examples in computational discourse theory are the theory of attentional/intentional structure (Grosz & Sidner 1986), and Rhetorical Structure Theory (Mann & Thompson 1987).

Grosz & Sidner (1986) present the basic elements of a computational theory of discourse structure. The two main questions the theory tries to answer are: What individuates a discourse? What makes it coherent? They claim the answers are intimately connected with two non-linguistic notions, attention and intention. Attention is an essential factor in explicating the processing of utterances in discourse. Intentions play a primary role in explaining discourse structure and defining discourse coherence. Grosz and Sidner claim that the intentions that underlie discourse are so diverse that approaches to discourse coherence based on selecting discourse relationships from a fixed set of alternative rhetorical patterns are unlikely to suffice. (See Hovy (1990) for a strong counterview.)

In this theory the linguistic structure consists of the discourse segments and an embedding relationship that can hold between them. The embedding relationships are a surface reflection of relationships among elements of the intentional structure. Linguistic expressions are among the primary indicators of discourse segment boundaries. The explicit use of certain words and phrases and more subtle cues, such as intonation or changes in tense and aspect, are included in the repertoire of linguistic devices that function to indicate these boundaries.

The attentional state is modeled by a set of focus spaces; changes in attentional state are modeled by a set of transition rules that specify the conditions for adding and deleting spaces. One focus space associated with each discourse segment. The focus space hierarchy is different/separate from the intentional (task) structure. Passonneau & Litman (1993), following Rotondo (1984), concede the difficulty of eliciting hierarchical intentional structure with any degree of consistency from their human judges. Not surprisingly, no fully implemented version of this theory exists.

Rhetorical Structure Theory (RST) (Mann & Thompson 1987) is a functionally-based descriptive tool for analysis of the rhetorical structure of text, designed to be used in automated systems. In RST, text is broken up into clausal units, each of which participates in a pairwise nucleus/satellite relationship. The pairs participate as components of larger pairwise units, building up a hierarchical discourse description. Some of the rhetorical relations linking the units are: elaboration, enablement, motivation, and background. The authors recognize that there are no reliable grammatical or lexical clues for automatically determining the structure, and often the relations can only be discerned by the underlying meaning of the text. The analysis is goal-oriented and might be less effective for texts that cannot be described well in this manner. RST has been used in generation systems, e.g., Moore & Pollack (1992).

Skorochod'ko's Topologies

Although many aspects of discourse analysis require a hierarchical model, in this work I choose to cast expository text into a linear sequence of segments, both for computational simplicity and because such a structure is appropriate for coarse-grained applications. This procedure is influenced by Skorochod'ko (1972), who suggests determining the semantic structure of a text (for the purposes of automatic abstracting) by analyzing it in terms of the topology formed by lexical interrelations found among its sentences.

Skorochod'ko (1972) suggests discovering a text's structure by dividing it up into sentences and seeing how much word overlap appears among the sentences. The overlap forms a kind of intra-structure; fully connected graphs might indicate dense discussions of a topic, while long spindly chains of connectivity might indicate a sequential account (see Figure 2.2). The central idea is that of defining the structure of a text as a function of the connectivity patterns of the terms that comprise it. This is in contrast with segmenting guided primarily by fine-grained discourse cues such as register change, focus shift, and cue words. From a computational viewpoint, deducing textual topic structure from lexical connectivity alone is appealing, both because it is easy to compute, and also because discourse cues are sometimes misleading with respect to the topic structure (Brown & Yule 1983)(§3).

In the Chained structure, each sentence describes a new situation or a new aspect of of the topic under discussion. Examples are chronological descriptions, where one event follows the next, and "road maps" in the beginning of technical papers outlining what the following sections contain. The Ringed structure is like the Chained structure except in the last portion of the discourse returns to what was initially discussed, perhaps as a summary discussion. The Monolith structure represents a densely interrelated discussion; each block contains references to terms in the other blocks, indicating several interwoven thematic threads. The Piecewise Monolithic structure consists of a sequence of dense interrelated discussions. Skorochod'ko did not define a hierarchical structure, perhaps because it is difficult to identify by using only term interrelations.

The topology most of interest to this work is the final one in the diagram, the Piecewise Monolithic Structure, since it represents sequences of densely interrelated discussions linked

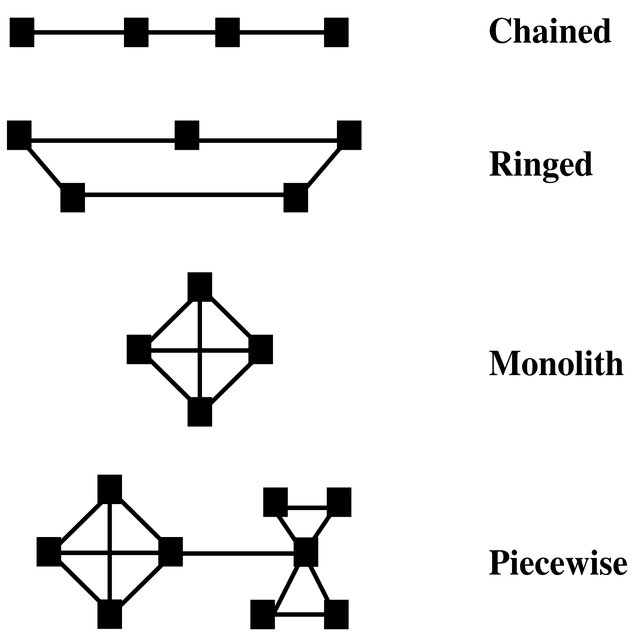


Figure 2.2: Skorochod'ko's text structure types. Nodes correspond to units of text such as sentences, and edges between nodes indicate strong term overlap between the text units. Correspondence between position of a node and position in the text depends on the kind of structure; this is described in more detail in the text.

together, one after another. This topology maps nicely onto that of viewing documents as a sequence of densely interrelated subtopical discussions, one following another. This assumption, as will be seen, is not always valid, but is nevertheless quite useful.

2.4.3 Grammars and Scripts

An alternative way of analyzing discourse structure is to propose a "grammatical" discourse theory. Many researchers have seen this as a natural extension to the ideas of sentence grammar. Fillmore (1981:147) makes a distinction between what a sentence grammarian does (looks for grammaticality and nongrammaticality) and what a discourse grammarian does (looks for sequiturity and nonsequiturity). Wilensky (1983b) also disputes the analogy between story grammars and sentence grammars, arguing that intuitions about stories are closer to our intuitions about the meanings of sentences than they are to our intuitions about sentences themselves.

Another alternative is to interpret texts from an artificial intelligence stance and try to fit the discourse into a predefined frame or script, e.g., Schank & Abelson (1977), Hahn (1990), DeJong (1982), Mauldin (1989). These approaches are usually used to create a summary of some kind. A variation on the theme is found in case-based reasoning, e.g., Kolodner (1983), Bareiss (1989), in which a discourse is adjusted to fit the expectations of a set of pre-analyzed discourses. The problem with this kind of approach is that it requires detailed knowledge about every domain that the analyzed texts discuss, and requires a very large amount of processing time for the analysis of only a few sentences; impractical requirements for a full-scale information access system.

2.5 Detecting Discourse Structure

Many different mechanisms have been proposed for the automated determination of discourse structure. Explicit cue words, (e.g., *now, well, so* in English (Schiffrin 1987)) are recognized as being meaningful cues, especially for spoken text. However, these cues are not unambiguous in usage, and considerable effort is required to determine the role of a particular instance of a cue (Hirschberg & Litman 1993). Other kinds of cues, such as tense (Webber 1987), (Hwang & Schubert 1992), are also informative but require a complex analysis. The next two subsections discuss two other means of determining discourse structure, making use of the patterns of cohesion indicators other than lexical cohesion, and lexical cohesion relations themselves.

2.5.1 Distributional Patterns of Cohesion Cues

Researchers have experimented with the display of patterns of cohesion indicators in discourse as an analytic device, for example, Grimes (1975)(Ch. 6) uses "span charts" to show the interaction of various thematic devices such as identification, setting and tense. Stoddard (1991) creates "cohesion maps" by assigning to each word a location on a two-dimensional grid corresponding to the word's position in the text (roughly, each sentence corresponds to a row), and then drawing a line between the location of a cohesive element and the location of its original referent. The resulting map looks somewhat like a column of hanging pine-needle bunches; thus texts can be compared visually for properties such as burstiness, density, and connection span. Each kind of cohesive element is assigned its own map, although for one example all three cohesion maps are superimposed. Here cohesion elements are pronominal referents, referents of definite articles, and verb agent displacements – lexical cohesion relations are not taken into account. Unfortunately, neither Stoddard nor Grimes analyze the resulting patterns or describe how to use them to segment or interpret the texts.

2.5.2 Lexical Cohesion Relations

The seminal linguistic work on lexical cohesion relations is that of Halliday & Hasan (1976). In a more abbreviated form, Raskin & Weiser (1987) point out that a distinction must be made between *cohesion* and *coherence* in a discourse. They state: "Coherence refers to the consistency of purpose, voice, content, style, form, and so on of a discourse as intended by the writer, achieved in the text, and perceived by the reader. Cohesion, on the other hand, is a textual quality which contributes to coherence through verbal cues" (p 48). One kind of cohesion cue is that of lexical cohesion, which "...results from the co-occurrence of semantically similar words that do not independently indicate cohesion" (p 204). Following Halliday & Hasan (1976), they describe two forms of lexical cohesion, *reiteration* and *collocation*, where the former refers to repetition of words or their synonyms, and the latter refers to terms that tend to co-locate in text, e.g., *night* and *day*, or *school* and *teacher*. Other kinds of cohesion cues relate to specific words that indicate particular relations, e.g., *afterwards* indicates a temporal relation between sentences, and *and* can indicate a conjunctive relationship. Relations such as anaphoric reference are considered to be grammatical cohesion, as opposed to lexical cohesion.

Phillips (1985) suggests "an analysis of the distribution of the selected text elements relative to each other in some suitable text interval ... for whatever patterns of association they may contract with each other as a function of repeated co-occurrence" (p 59). The resulting analysis leads to hypotheses of lexical meaning based on term co-occurrence, but the text structure elicited reflects not much beyond the chapter structure of the text books he investigates. Two other important approaches are those of Morris & Hirst (1991) and Youmans (1991), described in the following sections.

Morris and Hirst

Morris and Hirst's pioneering work on computing discourse structure from lexical relations (Morris & Hirst 1991; Morris 1988) is a precursor to the work reported on here. Morris, influenced by Halliday and Hasan's theory of lexical coherence (Halliday & Hasan 1976), developed an algorithm that finds chains of related terms via a comprehensive thesaurus (Roget's Fourth Edition). For example, the words *residential* and *apartment* both index the same thesaural category and can thus be considered to be in a coherence relation with one another. The chains are used to structure texts according to Grosz and Sidner's attentional/intentional theory of discourse structure (Grosz & Sidner 1986), and the extent of the chains correspond to the extent of a segment. The algorithm also incorporates the notion of "chain returns" – repetition of terms after a long hiatus – to close off an intention that spans over a digression.

Since the Morris and Hirst algorithm attempts to discover attentional/intentional structure, their goals are different than those of TextTiling. Specifically, the discourse structure they attempt to discover is hierarchical and more fine-grained than that discussed here. Morris (1988) provides five short example texts for which she has determined the intentional structure, and states that the lexical chains generated by her algorithm provide a good indication of the segment boundaries that Grosz and Sidner's theory assumes. In Morris (1988) and Morris & Hirst (1991), tables are presented showing the sentences spanned by the lexical chains and by the corresponding segments of the attentional/intentional structure (derived by hand). Figure 2.3 shows a graphical depiction of the same information for one of the test texts. It shows how different chains cover the structure at different levels of granularity, as well as which portions of the structure are not accounted for.

Several aspects of the algorithm are problematic, especially when applied to longer texts. First, the algorithm was executed by hand because the thesaurus is not generally available online. However, Project Gutenberg has donated an online copy of Roget's 1911 thesaurus which, although smaller and less structured than the thesaurus used by Morris, can be used for an implementation of the algorithm. Aside from the fact that using such a thesaurus lowers the quality of the connections found among terms, an implementation of the Morris algorithm using found that often the choice of which thesaural relation to use was not unambiguous.

Second, although ambiguous chain links were rare in Morris's texts, the texts analyzed here had many ambiguous links, even when connections were restricted to being made between terms in the same thesaurus category. Another problem results from the fact that the model does not take advantage of the tendency for multiple simultaneous chains might occur over the same intention. For example, Text 4-3 of Morris (1988) contains a discussion of the role of women in the USSR as embodied in the life of Raisa Gorbachev. Two different chains span most of the text: One consists of terms relating to the Soviet Union and the United States, and the other refers to women, men, husbands, and wives (see Figure 2.3).

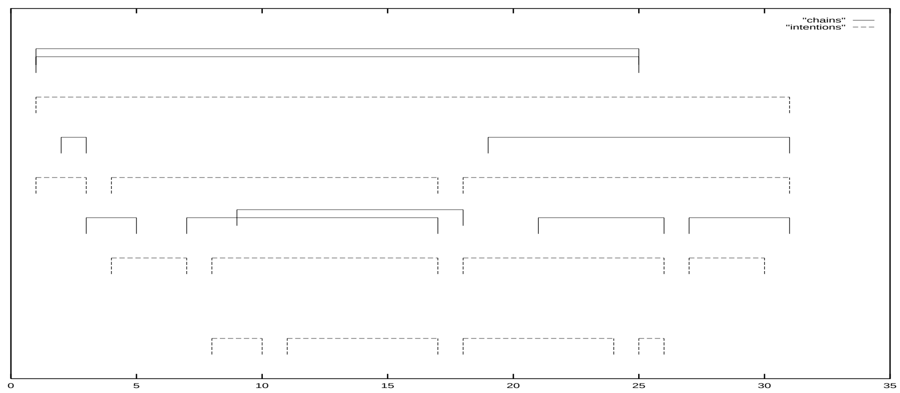


Figure 2.3: The target intentional structure and the extents of actual chains found in Morris 88 for text 4-3. The x-axis indicates sentence numbers, the y-axis indicates relative depth of embedding of the intentional structure.

Another, more serious problem arises when looking at longer texts: chain overlap. In other words, many chains end at a particular paragraph while at the same time many other chains extend past that paragraph. Figure 2.4 shows the distribution, by sentence number, of selected terms from the *Stargazers* text. The first two terms have fairly uniform distribution and so should not be expected to provide much information about the divisions of the discussion. The next two terms co-occur a few times at the beginning of the text (although *star* also occurs quite frequently at the end of the text as well), while terms *binary* through *planet* have considerable overlap from sentences 58 to 78. There is a somewhat well-demarked cluster of terms between sentences 35 and 50, corresponding to the grouping together of paragraphs 10, 11, and 12 by human judges who have read the text.

From the diagram it is evident that simply looking for chains of repeated terms is not sufficient for determining subtopic breaks. Even combining terms that are closely related semantically into single chains is insufficient, since often several different themes are active in the same segment. For example, sentences 37 - 51 contain dense interaction among the terms *move*, *continent*, *shoreline*, *time*, *species*, and *life*, and all but the latter occur only in this region. Few thesauri would group all of these terms together. However, it is the case that the interlinked terms of sentences 57 - 71 (*space*, *star*, *binary*, *trinary*, *astronomer*, *orbit*) are closely related semantically, assuming the appropriate senses of the terms have been determined.

One way to get around this difficulty is to extend the Morris algorithm to create graphs that plot the number of active chains against paragraph or sentence numbers. This option is discussed in more detail in Section 2.7.

Youmans

Another recent analytic technique that makes use of lexical information is described in Youmans (1991). Youmans introduces a variant on type/token curves, called the Vocabulary-Management Profile, that keeps track of how many first-time uses of terms occur at the midpoint of each 35-word window in a text. Youmans' goal is to study the distribution of vocabulary in discourse rather than to segment it along topical lines, but the peaks and valleys in the resulting plots "correlate closely to constituent boundaries and information flow" (although Youmans points out that they are correlated, but not directly related). Youmans begins with the hypothesis that new topics will be met with a sharp burst of new term uses, but this kind of activity is not visible on a typical type/token ratio plot. When instead of simple type/token ratios the number of new words within an interval of words are plotted, the changes become more visible.

Youmans discovers, upon examining many English narratives, essays, and transcripts, that new vocabulary is introduced less often in the first part than the second part of clauses and sentences, and that sharp upturns after deep valleys in the curve signal shifts to new subjects in essays and new episodes in stories. The analysis focuses on more fine-grained divisions than those of interest for TextTiling, subdividing each paragraph into multiple topic units. Youmans finds that for certain kinds of texts, the profile lags behind the onset of paragraphs for a sentence or two, since much expository writing includes repetition of information from one paragraph into the next,

Youmans also finds that longer intervals yield smoother plots, with lower peaks and

```
-------------------------------------------------------------------------------------------------------------
Sentence:        05   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80   85   90   95
-------------------------------------------------------------------------------------------------------------
14       form    1     111 1   1                               1 1     1   1       1     1     1    1
 8  scientist                   11            1   1               1          1      1  1
 5      space 11    1     1                                                             1
25       star    1             1                                       11 22  111112  1 1  1   11 1111     1
 5     binary                                                           11  1          1                    1
 4    trinary                                                            1   1         1                    1
 8 astronomer 1                1                                        1 1           1   1    1  1
 7      orbit   1                    1                                     12    1 1
 6       pull                          2     1 1                               1  1
16     planet   1    1       11               1           1                had 21  11111                  1     1
 7     galaxy    1                                                   1                1  11     1           1
 4      lunar            1  1     1       1
19       life 1  1  1                              1    11 1  11  1      1                 1 1    1 111  1 1
27       moon        13  1111   1 1 22 21  21    21           11 1
 3       move                                       1   1   1
 7  continent                                       2 1 1 2 1
 3  shoreline                                             12
 6       time                      1               1  1  1     1                                          1
 3      water                                 11            1
 6        say                                 1 1        1        11                1
 3    species                                       1  1  1
-------------------------------------------------------------------------------------------------------------
Sentence:        05   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80   85   90   95
-------------------------------------------------------------------------------------------------------------
```

Figure 2.4: Distribution of selected terms from the *Stargazer* text, with a single digit frequency per sentence number (blanks indicate a frequency of zero).

shallower valleys than with shorter intervals. Strongly influenced by linguistic notions, Youmans tries to cast the resulting peaks in terms of coordination and subordination relations, but in the discussion admits this does not seem like an appropriate use of the results. Youmans does not present an evaluation of how often the algorithm's valleys actually correspond to "information units", and leaves how to use the results to future work.

2.6 The TextTiling Algorithm

The TextTiling algorithm can be described in terms of a core and a collection of optional embellishments. In practice in experiments so far none of the embellishments significantly improve the performance of the core algorithm; this will be discussed in more detail below. I group the core algorithm and its variants together under the rubric of TextTiling.

Many researchers have studied the patterns of occurrence of characters, setting, time, and the other thematic factors, usually in the context of narrative. In contrast, TextTiling attempts to determine where a relatively large set of active themes changes simultaneously, regardless of the *type* of thematic factor. This is especially important in expository text in which the subject matter tends to structure the discourse more so than characters, setting, etc.[2] For example, in the *Stargazers* text, a discussion of continental movement, shoreline acreage, and habitability gives way to a discussion of binary and unary star systems. This is not so much a change in setting or character as a change in subject matter.

This theoretical stance bears a close resemblance to Chafe's notion of The Flow Model of discourse (Chafe 1979), in description of which he writes (pp 179-180):

> Our data ... suggest that as a speaker moves from focus to focus (or from thought to thought) there are certain points at which there may be a more or less radical change in space, time, character configuration, event structure, or, even, world. ... At points where all of these change in a maximal way, an episode boundary is strongly present. But often one or another will change considerably while others will change less radically, and all kinds of varied interactions between these several factors are possible.[3]

Although Chafe's work concerns narrative text, the same kind of observation applies to expository text. The TextTiling algorithms are designed to recognize episode boundaries by determining where the thematic components listed by Chafe change in a maximal way.

The TextTiling algorithms make use of lexical cohesion relations in a manner similar to that suggested by Skorochod'ko (1972) to recognize where the subtopic changes occur. This differs from the work of Morris & Hirst (1991) in several ways, the most important of which is that the algorithm emphasizes the interaction of multiple simultaneous themes,

[2]cf. Sibun (1992) for a discussion of how the form of people's descriptions often mirror the form of what they are describing.

[3]Interestingly, Chafe arrived at the Flow Model after working extensively with, and then becoming dissatisfied with, a Longacre-style hierarchical model of paragraph structure (Longacre 1979).

rather than following single threads of discussion alone. Main topics are themes that continue on throughout the ebb and flow of the interacting subtopics.

Many researchers (e.g., Halliday & Hasan (1976), Tannen (1989), Walker (1991)) have noted that term repetition is a strong cohesion indicator. In this work, term repetition alone, when used in terms of multiple simultaneous threads of information, is a very useful indicator of subtopic structure. This section describes the core algorithm for discovering subtopic structure using term repetition as a lexical cohesion indicator.

The core algorithm compares, for a given window size, each pair of adjacent blocks of text according to how similar they are lexically (see Figure 2.5). This method assumes that the more similar two blocks of text are, the more likely it is that the current subtopic continues, and, conversely, if two adjacent blocks of text are dissimilar, the current subtopic gives way to a new one.

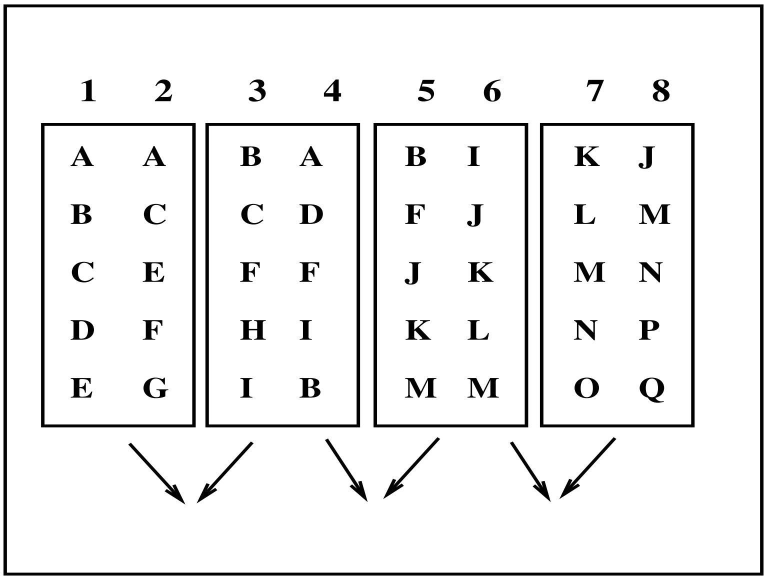


Figure 2.5: Illustration of the core lexical cohesion comparison algorithm. Letters signify lexical items, numbers signify sentence numbers. In the diagram, similarity comparison is done on adjacent blocks with a blocksize of 2. Arrows indicate which blocks are compared to yield scores for sentence gaps 2, 4, and 6. Blocks are shifted by one sentence for similarity measurements for gaps 3, 5, and 7.

The rationale behind this strategy is that it is an attempt to detect when a dense, interrelated discussion ends and a new one begins, in the spirit of Skorodch'ko's Piecewise Monolithic discourse topology. The appearance of a set of new terms indicates the onset of a new topic, as in Youmans' approach, but the repetition of existing terms also provides helpful evidence – that is, evidence that the current discussion is still ongoing. However, there is no explicit requirement about how close together individual terms must be. In other words, the algorithm does not need to specify how far apart individual terms can be; rather it looks for a change in the overall patterns among the terms in the blocks being compared.

The core algorithm has three main parts:

1. Tokenization
2. Similarity Determination
3. Boundary Identification

Each is described in detail below.

2.6.1 Tokenization

Tokenization refers to the division of the input text into individual lexical units, and is sensitive to the format of the input text. For example, if the document has markup information, the header and other auxiliary information is skipped until the body of the text is located. Tokens that appear in the body of the text are converted to all lower-case characters and checked against a "stoplist" of 898 words, the most frequent terms in a large text collection. If the token is a stopword then it is not passed on to the next step. Otherwise, the token is reduced to its root by a morphological analysis function which uses WordNet's noun and verb term lists and exception lists, converting regularly and irregularly inflected nouns and verbs to their roots.

The text is subdivided into psuedosentences of a pre-defined size *w* (a parameter of the algorithm) rather than actual syntactically-determined sentences, thus circumventing normalization problems. For the purposes of the rest of the discussion these groupings of tokens will be referred to as *token-sequences*. In practice, setting *w* to 20 tokens per token-sequence works best for many texts. The morphologically-analyzed token is stored in a table along with a record of the token-sequence number it occurred in, and how frequently it appeared in the token-sequence. A record is also kept of the locations of the paragraph breaks within the text.

2.6.2 Similarity Determination

The next step is the comparison of adjacent pairs of blocks of token-sequences for overall lexical similarity. (See the sketch in Figure 2.5.) Another important parameter for the algorithm is the *blocksize*: the number of token-sequences that are grouped together into a block to be compared against an adjacent group of token-sequences. This value, labeled k, varies slightly from text to text; as a heuristic it is the average paragraph length (in token-sequences). In practice, a value of $k = 6$ works well for many texts. Actual paragraphs are not used because their lengths can be highly irregular, leading to unbalanced comparisons.

Similarity values are computed for every token-sequence gap number; that is, a score is assigned to token-sequence gap i corresponding to how similar the token-sequences from token-sequence $i - k$ through i are to the token-sequences from $i + 1$ to $i + k + 1$. Note that this moving window approach means that each token-sequence appears in $k * 2$ similarity computations.

Similarity between blocks is calculated by a cosine measure: given two text blocks b_1 and b_2, each with k token-sequences,

$$sim(b_1, b_2) = \frac{\sum_t w_{t,b_1} w_{t,b_2}}{\sqrt{\sum_t w_{t,b_1}^2 \sum_{t=1}^n w_{t,b_2}^2}}$$

where t ranges over all the terms that have been registered during the tokenization step, and w_{t,b_1} is the weight assigned to term t in block b_1. In the core version of the algorithm, the weights on the terms are simply their frequency within the block. Thus if the similarity score between two blocks is high, then the blocks have many terms in common. This formula yields a score between 0 and 1, inclusive.

These scores can be plotted, token-sequence number against similarity score. However, since similarity is measured between blocks b_1 and b_2, where b_1 spans token-sequences $i - k$ through i and b_2 spans $i + 1$ to $i + k + 1$, the measurement's x-axis coordinate falls between token-sequences i and $i + 1$. Therefore, the x-axis corresponds to token-sequence *gap* number i.

2.6.3 Boundary Identification

Boundary identification takes place in several steps. First, the plot is smoothed with average smoothing; that is,

for each token-sequence gap g and an even window size $w + 1$
    find the scores of the $w/2$ gaps to the left of g
    find the scores of the $w/2$ gaps to the right of g
    find the score at g
    take the average of these scores and assign it to g'
repeat this procedure n times

In practice, for most of the examined texts, one round of average smoothing with a window size of three works best.

Boundaries are determined by changes in the sequence of similarity scores. The token-sequence gap numbers are ordered according to how steeply the slopes of the plot are to either side of the token-sequence gap, rather than by their absolute similarity score. For a given token-sequence gap i, the algorithm looks at the scores of the token-sequence gaps to the left of i as long are their values are increasing. When the values to the left peak out, the difference between the score at the peak and the score at i is recorded. The same procedure takes place with the token-sequence gaps to the right of i; their scores are examined as long as they continue to rise. The relative height of the peak to the right of i is added to the relative height of the peak to the left. (A gap occurring at a peak will have a score of zero since neither of its neighbors is higher than it.)

These new scores, called depth scores, corresponding to how sharp a change occurs on both sides of the token-sequence gap, are sorted. Segment boundaries are assigned to the token-sequence gaps with the largest corresponding scores, adjusted as necessary to correspond to true paragraph breaks. A proviso check is done that prevents assignment of very close adjacent segment boundaries. Currently there must be at least three intervening token-sequences between boundaries. This helps control for the fact that many texts have spurious header information and single-sentence paragraphs.

A consequence of the boundary determination strategy is that a token-sequence gap that lies between two sharply rising peaks will receive a higher score than a token-sequence gap in the middle of a long valley with low hills. Thus a gap with a high peak on only one side can receive a good-sized score. A potential problem occurs if there is a rise on one side of a gap, and a decline on the other. However, the gap at the bottom of the decline will receive an even larger score than the first gap and so will overrule the first gap's score, if the two gaps are close together. On the other hand if the two gaps are far apart, there is probably a call for the intermediate gap to serve as a boundary.

Another issue concerns the number of segments to be assigned to a document. Every paragraph is a potential segment boundary. Any attempt to make an absolute cutoff is problematic since there would need to be some correspondence to the document style and length. A cutoff based on a particular valley depth is similarly problematic.

I have devised a method for determining how many boundaries to assign that scales with the size of the document and is sensitive to the patterns of similarity scores that it produces. The cutoff is a function of the average and standard deviations of the depth scores for the text under analysis. Currently a boundary is drawn only if the depth score exceeds $\bar{s} - \sigma/2$.

2.6.4 Embellishments

There are several ways to modify the algorithm in order to attempt to improve its results. Some of these are:

- Varying the specifics of tokenization, e.g., increasing or reducing the stoplist or the degree morphological analysis (e.g., derivational vs. inflectional vs. no analysis)
- Using thesaural relations in addition to term repetition to make better estimates about the cohesiveness of the discussion.
- Using localized discourse cue information to help better determine exact locations of boundaries.
- Weighting terms according to their prior probability, how frequent they are in the text under analysis, or some other property.

- Using a different similarity measure, such as one that weights the terms according to a gaussian distribution centered at each token-sequence gap number.
- Treating the plot as a probabilistic time series and detected the boundaries based on the likelihood of a transition from nontopic to topic.[4]

Earlier work (Hearst 1993) incorporated thesaural information into the algorithms; surprisingly the latest experiments find that this information degrades the performance. This could very well be due to problems with the thesaurus and assignment algorithm used (a variation on that described in Chapter 4. A simple algorithm that just posits relations among terms that are a small distance apart according to WordNet (Miller *et al.* 1990) or Roget's 1911 thesaurus (from Project Gutenberg), modeled after Morris and Hirst's heuristics, might work better. Therefore I do not feel the issue is closed, and instead consider successful grouping of related words as future work. As another possible alternative Kozima (1993) has suggested using a (computationally expensive) semantic similarity metric to find similarity among terms within a small window of text (5 to 7 words). This work does not incorporate the notion of multiple simultaneous themes but instead just tries to find breaks in semantic similarity among a small number of terms. A good strategy may be to substitute this kind of similarity information for term repetition in algorithms like those described here. Another possibility would be to use semantic similarity information as computed in Schütze (1993b), Resnik (1993), or Dagan *et al.* (1993).

The use of discourse cues for detection of segment boundaries and other discourse purposes has been extensively researched, although predominantly on spoken text (see Hirschberg & Litman (1993) for a summary of six research groups' treatments of 64 cue words). It is possible that incorporation of such information may help improve the cases where the algorithm is off by one paragraph, as might reference resolution or an account of tense and aspect. Informal experiments with versions of all of the other items do not seem to produce significantly better results than the most stripped-down version of the core algorithm.

Another way to alter the algorithm is to change the comparison strategy. It is possible to modify the approach of Morris & Hirst (1991), discussed above, to take multiple simultaneous themes into account, and to apply it to the multi-paragraph segmentation problem as opposed to the attentional/intentional segment recognition problem. Rather than assuming that each chain corresponds directly to one segment, and vice versa, an algorithm can create a collection of active chains, and then place boundaries at the points in the text where more chains are inactive than active (see Figure 2.6). This approach does not make use of explicit chain returns; they are accounted for implicitly instead. A version of Youmans' algorithm (Youmans 1991), also discussed above, and modified to apply to larger segmentation units, might also prove successful, although preliminary experiments did not show it to perform significantly better.

---

[4] I am grateful to Isabelle Guyon for her help with this suggestion.

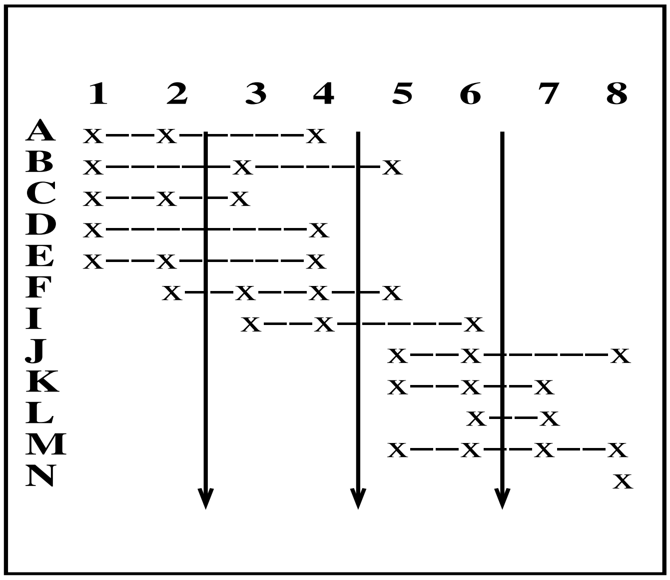


Figure 2.6: Accumulating counts of chains of terms: letters signify lexical items, numbers signify token-sequence numbers, ‘x’ indicates that the term occurs in the token-sequence, ‘-’ indicates continuation of a chain, and arrows cut through the active chains that contribute to the cumulative count for token-sequence gaps 2, 4, and 6. In the diagram there is evidence for a break between token-sequences 4 and 5 because there are few active chains there.

2.7 Evaluation

One way to evaluate these segmentation algorithms is to compare against judgments made by human readers, another is to see how well the results improve a computational task, and a third possible evaluation measure is to compare the algorithms against texts pre-marked by authors. This section compares the algorithm against reader judgments, since author markups are fallible and are usually applied to text types that this algorithm is not designed for, and Chapter 3 shows how to use tiles in a task (although it does not formally prove that the results of the algorithm improve the task more than some other algorithm with similar goals would).

2.7.1 Reader Judgments

Judgments were obtained from seven readers for each of thirteen magazine articles which satisfied the length criteria (between 1800 and 2500 words)[5] and which contained little structural demarkation. The judges were asked simply to mark the paragraph boundaries at

[5]One longer text of 2932 words was used since reader judgments had been obtained for it from an earlier experiment. Note that this represents an amount of test data on the order of that used in the experiments of Passonneau & Litman (1993). Judges were technical researchers. Two texts had three or four short headers which we removed.

which the topic changed; they were not given more explicit instructions about the granularity of the segmentation.

Figure 2.7(a) shows the boundaries marked by seven judges on the *Stargazers* text. This format helps illuminate the general trends made by the judges and also helps show where and how often they disagree. For instance, all but one judge marked a boundary between paragraphs 2 and 3. The dissenting judge did mark a boundary after 3, as did two of the concurring judges. The next three major boundaries occur after paragraphs 5, 9, 12, and 13. There is some contention in the later paragraphs; three readers marked both 16 and 18, two marked 18 alone, and two marked 17 alone. The outline in Section 2.1 gives an idea of what each segment is about.

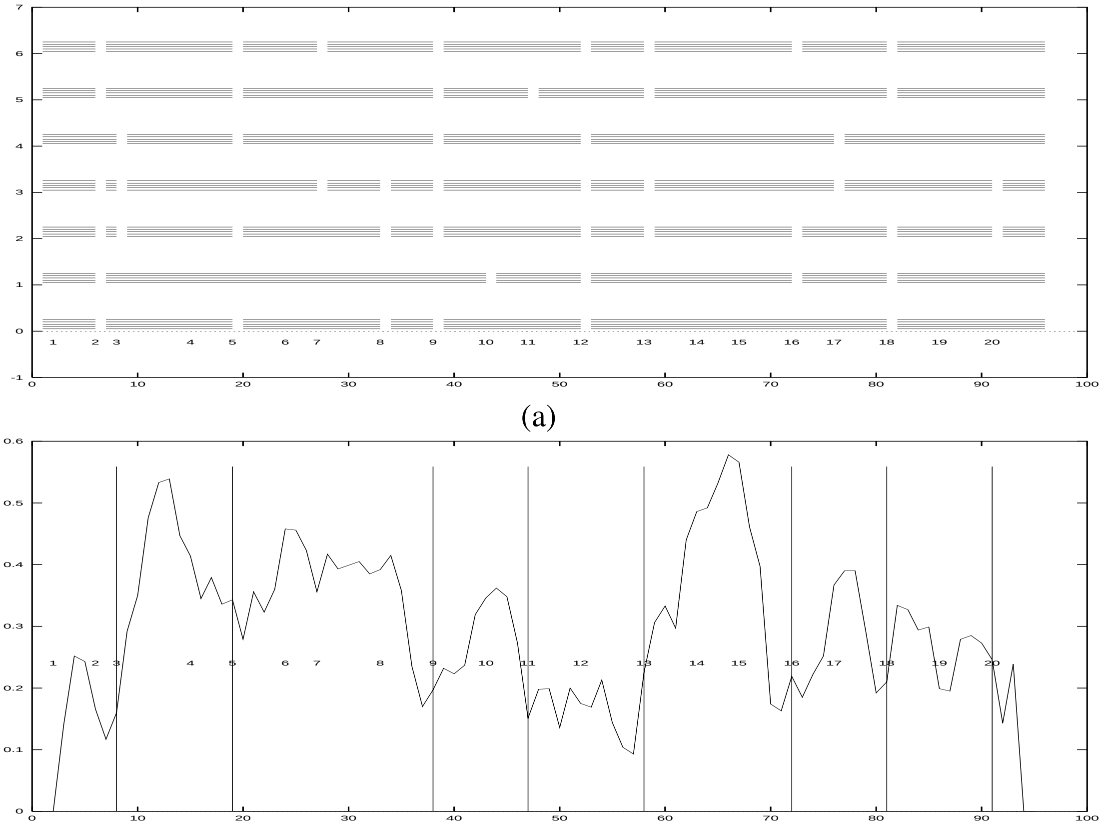
Passonneau & Litman (1993) discuss at length considerations about evaluating segmentation algorithms according to reader judgment information. As Figure 2.7(b) shows, agreement among judges is imperfect, but trends can be discerned. In Passonneau & Litman's (1993) data, if 4 or more out of 7 judges mark a boundary, the segmentation is found to be significant using a variation of the Q-test (Cochran 1950). My data showed similar results. However, it isn't clear how useful this significance information is, since a simple majority does not provide overwhelming proof about the objective reality of the subtopic break. Since readers often disagree about where to draw a boundary marking for a topic shift, one can only use the general trends as a basis from which to compare different algorithms. Since the goals of TextTiling are better served by algorithms that produce more rather than fewer boundaries, I set the cutoff for "true" boundaries to three rather than four judges per paragraph.[6] The remaining gaps are considered nonboundaries.

2.7.2 Results

Figure 2.7(b) shows a plot of the results of applying the block comparison algorithm to the *Stargazer* text. When the lowermost portion of a valley is not located at a paragraph gap, the judgment is moved to the nearest paragraph gap.[7] For the most part, the regions of strong similarity correspond to the regions of strong agreement among the readers. (These results were fifth highest out of the 13 test texts.) Note however, that the similarity information around paragraph 12 is weak. This paragraph acts as a summary paragraph, summarizing the contents of the previous three and revisiting much of the terminology that occurred in them all in one location (in the spirit of a Grosz & Sidner (1986) "pop" operation). Thus it displays low similarity both to itself and to its neighbors. This is an example of a breakdown caused by the assumption about the linear sequence of the subtopic discussions. It is possible that an additional pass through the text could be used to find structure of this kind.

[6] Paragraphs of three or fewer sentences were combined with their neighbor if that neighbor was deemed to follow at "true" boundary, as in paragraphs 2 and 3 of the *Stargazers* text.

[7] The need for this adjustment might be explained in part by Stark (1988) who shows that readers disagree measurably about where to place paragraph boundaries when presented with texts with those boundaries removed.



(a)

(b)

Figure 2.7: (a) Judgments of seven readers on the *Stargazer* text. Internal numbers indicate location of gaps between paragraphs; x-axis indicates token-sequence gap number, y-axis indicates judge number, a break in a horizontal line indicates a judge-specified segment break. (b) Results of the block similarity algorithm on the *Stargazer* text. Internal numbers indicate paragraph numbers, x-axis indicates token-sequence gap number, y-axis indicates similarity between blocks centered at the corresponding token-sequence gap. Vertical lines indicate boundaries chosen by the algorithm; for example, the leftmost vertical line represents a boundary after paragraph 3. Note how these align with the boundary gaps of (a).

	Precision		Recall	
	avg	sd	avg	sd
Baseline 33%	.44	.08	.37	.04
Baseline 41%	.43	.08	.42	.03
Chains	.64	.17	.58	.17
Blocks	.66	.18	.61	.13
Judges	.81	.06	.71	.06

Table 2.1: Precision and Recall values for 13 test texts.

	Total	Baseline 41% (avg)				Blocks				Chains				Judges (avg)			
Text	Possible	Prec	Rec	C	I	Prec	Rec	C	I	Prec	Rec	C	I	Prec	Rec	C	I
1	9	.44	.44	4	5	1.0	.78	7	0	1.0	.78	7	0	.78	.78	7	2
2	9	.50	.44	4	4	.88	.78	7	1	.75	.33	3	1	.88	.78	7	1
3	9	.40	.44	4	6	.78	.78	7	2	.56	.56	5	4	.75	.67	6	2
4	12	.63	.42	5	3	.86	.50	6	1	.56	.42	5	4	.91	.83	10	1
5	8	.43	.38	3	4	.70	.75	6	2	.86	.75	6	1	.86	.75	6	1
6	8	.40	.38	3	9	.60	.75	6	3	.42	.63	5	8	.75	.75	6	2
7	9	.36	.44	4	7	.60	.56	5	3	.40	.44	4	6	.75	.67	6	2
8	8	.43	.38	3	4	.50	.63	5	4	.67	.75	6	3	.86	.75	6	1
9	9	.36	.44	4	7	.50	.44	4	3	.60	.33	3	2	.75	.67	6	2
10	8	.50	.38	3	3	.50	.50	4	3	.63	.63	5	3	.86	.75	6	1
11	9	.36	.44	4	7	.50	.44	4	4	.71	.56	5	2	.75	.67	6	2
12	9	.44	.44	4	5	.50	.56	5	5	.54	.78	7	6	.86	.67	6	1
13	10	.36	.40	4	7	.30	.50	5	9	.60	.60	6	4	.78	.70	7	2

Table 2.2: Scores by text, showing precision and recall. (C) indicates the number of correctly placed boundaries, (I) indicates the number of inserted boundaries. The number of deleted boundaries can be determined by subtracting (C) from Total Possible.

The final paragraph is a summary of the entire text; the algorithm recognizes the change in terminology from the preceding paragraphs and marks a boundary; only two of the readers chose to differentiate the summary; for this reason the algorithm is judged to have made an error even though this sectioning decision is reasonable. This illustrates the inherent fallibility of testing against reader judgments, although in part this is because the judges were given loose constraints.

Following the advice of Gale *et al.* (1992a), I compare the algorithm against both upper and lower bounds. The upper bound in this case is the averages of the reader judgment data. The lower bound is a baseline algorithm that is a simple, reasonable approach to the problem that can be automated. In the test data, boundaries are placed in about 41% of the paragraph gaps. A simple way to segment the texts is to place boundaries randomly in the document, constraining the number of boundaries to equal that of the average number of paragraph gaps assigned by judges. A program was written that places a boundary randomly at each potential gap 41% of the time, was run a large number of times (10,000) for each text, and the average of the scores of these runs was found.

The algorithms are evaluated according to how many true boundaries they select out of the total selected (precision) and how many true boundaries are found out of the total possible (recall) (Salton 1988). The recall measure implicitly signals the number of missed boundaries (false negatives, or deletion errors); the table also indicates the number of false positives, or insertion errors, explicitly. The precision and recall for the average of the results appear in Table 2.1 (results at 33% are also shown for comparison purposes).

I also compared the core TextTiling algorithm against the chaining algorithm variant discussed in Section 2.6.4. The best variation on the chaining algorithm allows gaps of up to six token-sequences before the chain is considered to be broken. For both algorithms, w is 20, and morphological analysis and a stoplist are applied, as described in Section 2.6.1.

Table 2.1 shows that the blocking algorithm is sandwiched between the upper and lower bounds. The block similarity algorithm seems to work slightly better than the chaining algorithm, although the difference may not prove significant over the long run. Table 2.2 shows some of these results in more detail.

In many cases the algorithms are almost correct but off by one paragraph, especially in the texts that the algorithm performs poorly on. When the block similarity algorithm is allowed to be off by one paragraph, there is dramatic improvement in the scores for the texts that lower part of Table 2.2, yielding an overall precision of 83% and recall of 78%. As in Figure 2.7, it is often the case that where the algorithm is incorrect, e.g., paragraph gap 11, the overall blocking is very close to what the judges intended.

2.8 An Extended Example: The Tocqueville Chapter

This section illustrates the results of TextTiling on Chapter 1, Volume 1 of Tocqueville's *Democracy in America* discussed in Section 2.3.3. As mentioned there, this text is interesting because the author has provided a subtopic-like structure in the chapter preamble. The text of the chapter, labeled with paragraph numbers and sectioning information from the tiling algorithm, appears in Appendix A. The paragraph-level breakdown of the subtopic descriptions is reproduced in Figure 2.8 for convenient reference and Figure 2.9 shows the corresponding plot produced by the TextTiling algorithm. Note that the last two paragraphs in the text are summary in nature, and are not referred to in the subtopic list.

Comparing the results of tiling against the subtopic list of Figure 2.8, we see that the algorithm is generally successful. However, it does make some off-by-one errors and inserts at least one boundary that is not specified by the subtopic list. Figure 2.10 compares the results of the algorithm to that specified in Tocqueville's subtopic list according to token-sequence gap number (the final paragraphs are not shown since they are not referred to in Tocqueville's subtopic list). Using the precision/recall measures of the previous section we see that according to these boundaries the algorithm correctly chooses 6/9 of the possible boundaries (recall = 67%), and of the boundaries it chooses, 6/9 were also chosen according to the subtopic structure (precision = 67%). Looking at Figure 2.10 and at the text of the

01-06 North America divided into two vast regions, one inclining towards the Pole, the other towards the Equator
07-09 Valley of the Mississippi
10-11 Traces found there of the revolutions of the globe
12-13 Shore of the Atlantic Ocean, on which the English colonies were founded
14-16 Different aspects of North and of South America at the time of their discovery
17-18 Forests of North America
19-19 Prairies
20-20 [The tribes'] outward appearance, customs, and languages
21-25 Wandering tribes of natives
26-28 Traces of an unknown people.

Figure 2.8: Paragraph-level breakdown of the subtopic structure of Tocqueville Ch. 1 Vol. 1, repeated here for convenient reference.

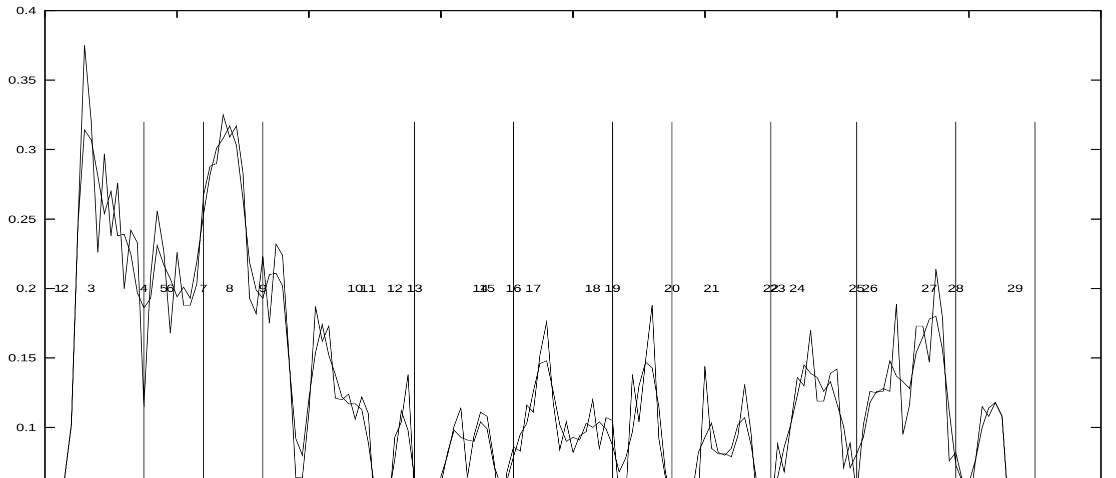


Figure 2.9: Results of the block similarity algorithm on Chapter 1, Volume 1 of *Democracy in America*. Internal numbers indicate paragraph gap numbers (e.g., the number '10' indicates that the boundary falls between paragraphs 9 and 10), x-axis indicates token-sequence gap number, y-axis indicates similarity between blocks centered at the corresponding token-sequence gap. Vertical lines indicate boundaries chosen by the algorithm.

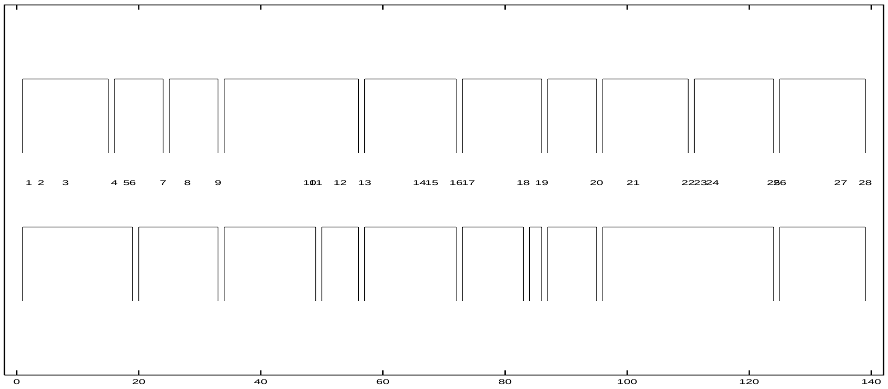


Figure 2.10: Another view of the results of the block TextTiling algorithm on the Tocqueville chapter. The bottom row corresponds to an interpretation of Tocqueville's subtopic labels, the top row corresponds to the output of the algorithm. The internal numbers indicate paragraph gap numbers, and the x-axis corresponds to token-sequence gap number.

chapter, we see that the results are better than these numbers might indicate.

For example, since there is a mention of *prairies* in the subtopic list, I have chosen to specify a break between paragraphs 19 and 20, despite the fact that paragraph 19 is a continuation of the discussion of forests and has only the barest mention of prairies. The algorithm produces a healthy peak corresponding to the focus on woodlands and flora of paragraphs 17 - 19. The stretch of paragraphs 21 - 25 is broken into two peaks by the algorithm, the first corresponding to a discussion of the characteristics of a people, and the second corresponding to a comparison between Europeans and these people.

The discussion corresponding to "Valley of the Mississippi" was assigned paragraphs 7 - 9, although most of the discussion, with the exception of the first sentence of paragraph 7, refers to the river more than to the valley. Correspondingly, the plot in Figure 2.9 rises midway through the discussion of paragraph 7 and the program has to make a choice between marking the boundary following paragraph 6 or paragraph 7. Since neither one corresponds directly to the valley in the plot, the decision goes to gap with the sharper rise on one side.

Another example of the content of the paragraphs not corresponding to their form, I've marked paragraphs 10 and 11 as corresponding to "Traces found there [in the Valley of the Mississippi] of the revolutions of the globe". However, the discussion of the river continues about one third of the way through paragraph 10, after which the discussion of the primeval ocean starts up. This pattern is reflected in the plot of Figure 2.9.

Finally, the algorithm does not mark a boundary between paragraphs 11 and 12. There is a dip in the plot following paragraph 12 (which is off by one sentence from the desired boundary, after 11), but the restriction on allowing very close neighbors prevents this from being marked, due to paragraph 12's proximity to 13.

Overall, then, the algorithm does quite well at identifying the main subtopic boundaries of the Tocqueville chapter. In several cases in which the algorithm seems to be off, it is the result of the fact that the actual transition takes place mid-paragraph. This is perhaps an argument for loosening the restriction of TextTiling into non-overlapping text units, especially when used for the purposes of user interface display.[8]

2.9 Conclusions

This chapter has described algorithms for the segmentation of expository texts into multi-paragraph discourse units that reflect the subtopic structure of the texts. It has introduced the notion of the recognition of multiple simultaneous themes as an approximation to Skorodch'ko's Piecewise Monolithic text structure type. The algorithm is fully implemented and term repetition alone, without use of thesaural relations, knowledge bases, or inference mechanisms, works well for many of the experimental texts.

The chaining algorithm variation is adapted from that of Morris & Hirst (1991), with the following differences: (i) the scores from multiple simultaneous chains are combined at the boundary of each sentence (or token-sequence) and used to determine where segment breaks should be made, (ii) no thesaurus terms are used, and (iii) no chain returns are used to determine if a chain that broke off restarted later. This algorithm seems comparable to the block algorithm; in both cases, one algorithm performs better than the other on some of the test texts. This may well occur because both algorithms make use only of lexical co-occurrence information, and the evidence for boundaries given by this kind of information is impoverished compared to the phenomena it tries to account for. Furthermore, the reader judgment data being used as a yardstick is not terribly reliable since agreement among the judges, although significant at frequency four according to the measure of Passonneau & Litman (1993), is still rather low. Apparently there is more than one way to tile a text, as indicated by disagreement among judges and algorithms. Furthermore, in both versions of the algorithm, changes to the parameters of the algorithm perturb the resulting boundary markings. This is an undesirable property and perhaps could be remedied with some kind of information-theoretic formulation of the problem.[9]

These issues are not too damaging if the results are useful. Chapter 3 describes a new information access framework which uses the results of the block tiling algorithm to determine whether terms in a query overlap in a passage. Although no attempt is made there to show formally that the tiles perform better than randomly divided texts (since platforms for evaluation of such information do not currently exist), informal interactions with that system indicate that when tiling is correct the results of the system are better than when tiling is incorrect. This indirect evidence implies that the technique, despite the disagreement in judgments among readers and the errors in the algorithm itself, is better than arbitrarily divided texts or paragraphs alone.

[8] I am grateful to Jan Pedersen for this observation.

[9] This idea was suggested by Graeme Hirst and Andreas Stolcke.

Chapter 3

Term Distribution in Full-Text Information Access

3.1 Introduction

As mentioned in Chapter 1, most information retrieval methods are better suited to titles and abstracts than full text documents. In this chapter, I argue that the advent of full-length text should be accompanied by corresponding new approaches to information access. Most importantly, I emphasize that even more than short text, *full text requires context*: term context is important in computing retrieval rankings and in displaying retrieved passages and documents.

Information access mechanisms should not be thought of as retrieval in isolation. The mechanisms for querying as well as display are intimately tied with the retrieval mechanism, whether the implementor recognizes this or not. Cutting *et al.* (1990:1) advocate a text access paradigm that "weaves together interface, presentation and search in a mutually reinforcing fashion"; this viewpoint is adopted here as well.

In Hearst & Plaunt (1993), we suggest that in the analysis of full-length texts a distinction should be made between main topics and subtopics, and we suggest that users be allowed to specify a search for a subtopic *with respect to* some main topic. To see why this distinction might be useful, consider the following scenario: A user would like to find a discussion of funding for cold fusion research. There is a long text about cold fusion that has a two-paragraph discussion of funding two-thirds of the way in. This discussion, because it is in the context of a document about cold fusion, does not mention the term *cold fusion* anywhere near the discussion of funding. A full-document retrieval will either assign low rank to this document because funding-related terms are infrequent relative to the whole, or else it will assign high rank to *any* articles about cold fusion. A retrieve against individual paragraphs or segments will either assign low rank to this document because it will see only funding terms but no cold fusion terms in the relevant segment, or it will give high rank to *any* documents that have discussions of funding. Thus the distribution of terms with respect to one another can play a role in determining the potential relevance of a document to a query.

In this chapter I emphasize the importance of relative term distribution information in information access from full-text documents. The chapter first discusses the standard information retrieval ranking measures. It then suggests that because the makeup of long texts is qualitatively different from that of abstracts and short texts, the standard approaches are not necessarily appropriate for long texts. Since a critical aspect of long text structure is the pattern of term distribution, I enumerate the possible distribution relations that can hold between two sets of terms, and make predictions about the usefulness of each distribution type.

I then point out that existing approaches to information access do not suggest a way to use this distributional information. Furthermore, standard ranking mechanisms are opaque; users do not know what role their query terms played in the ranking of the retrieved documents. This problem is exacerbated when retrieving against full-text documents, since it is less clear how the terms in the query relate to the contents of a long text than an abstract.

An analogous situation arises in the use of query languages: in both cases the situation can be improved by making information visible and explicit to the largest extent possible (while avoiding complexity). A serious attitude toward considerations of clarity and conciseness leads to an information access paradigm in which the query specification and the results of retrieval are integrated, and the relationships between the query and the retrieved documents are displayed clearly.

Toward these ends, I introduce a new display paradigm, called TileBars, which allows the user to simultaneously view the relative length of the retrieved documents, the relative frequency of the query terms, and their distributional properties with respect to the document and each other. I show TileBars to be a useful analytical tool for determining document relevance when applied to sample queries from the TREC collection (Harman 1993), and I suggest using this tool to help explain why standard information retrieval measures succeed or fail for a given query.

I also discuss general issues in passage retrieval. No test collections exist for passage retrieval, and in general the issue has not been well-defined. Therefore, I suggest that the issues of relative distribution of terms and context from which the passage is extracted be taken into account in the development of a test collection for passage retrieval.

3.2 Background: Standard Retrieval Techniques

The purpose of information retrieval is to develop techniques to provide effective access to large collections of objects (containing primarily text) with the purpose of satisfying a user's stated information need (Croft & Turtle 1992). The most common approaches for this purpose are Boolean term retrieval and similarity search. I use the term "similarity search" as an umbrella term covering the vector space model (Salton 1988), probabilistic models (van Rijsbergen 1979), (Cooper *et al.* 1994), (Fuhr & Buckley 1993), and any other approach which attempts to find documents that are most similar to a query or to one another based on the terms they contain. In similarity search, the best overall matches are not necessarily the ones in which the largest percentage of the query terms are found, however. For example, given a query with 30 terms in it, the vector space model permits a document that contains only a few of the query terms to be ranked very highly if these words occur infrequently in the corpus as a whole but frequently in the document.

In the vector space model (Salton 1988), a query's terms are weighted and placed into a vector that is compared against vectors representing the documents of the collection. The underlying assumption is that documents' content can be represented in a geometric space and the relative distance between their vectors represents their relative semantic distance. In probabilistic models (van Rijsbergen 1979), the goal is to rank the database of documents in order of their probability of usefulness for satisfying the user's stated information need. However, in practice these systems also represent queries and documents with weighted terms and try to predict the probability of relevance of a document to a query by combining the scores of the weighted terms.

In Boolean retrieval a query is stated in terms of disjunctions, conjunctions, and negations among sets of documents that contain particular words and phrases. Documents are retrieved whose contents satisfy the conditions of the Boolean statement. The users can have more control over what terms actually appear in the retrieved documents than they do with similarity search. However, a drawback of Boolean retrieval is that in this framework no ranking order is specified. This problem is sometimes assuaged by applying ranking criteria as used in similarity search to the results of the Boolean search (Fox & Koll 1988).

Most information retrieval similarity measures treat the terms in a document uniformly throughout. That is, a term's weight is the same no matter where it occurs in the text.[1] Many researchers assume this is a valid assumption when working with abstracts, since it is a fair approximation to say that the location of the term does not significantly effect its import. These comments apply as well to short news articles, another text type commonly studied in information retrieval research.

Although there are other approaches, such as knowledge-based systems, e.g., McCune *et al.* (1985),Fung *et al.* (1990), Mauldin (1991), DeJong (1982), which attempt to interpret the text to some degree, and systems that attempt to answer questions, e.g., O'Connor (1980) and Kupiec (1993), the bulk of information retrieval research has focused on satisfying a query that can be paraphrased as: "Find more documents like this one." This a natural way to phrase a query, and is perhaps one of the more accessible to formalization, but it is certainly not the only useful question to allow a user to ask. In the next section I describe why alternatives to the query "Find more documents like this one" should be considered for full-text information access, and outline an alternative viewpoint on how to retrieve and display information from full-text documents.

[1] Small windows of adjacency information are sometimes used in Boolean systems, but not in probabilistic or vector-space models. The recent experiments of Keen (1991),Keen (1992) are an exception to this.

3.3 Long Texts and Their Properties

A problem with applying traditional information retrieval methods to full-length text documents is that the structure of full-length documents is quite different from that of abstracts. Abstracts are compact and information-dense. Most of the (non-closed-class) terms in an abstract are salient for retrieval purposes because they act as placeholders for multiple occurrences of those terms in the original text, and because generally these terms pertain to the most important topics in the text. Consequently, if the text is of any sizeable length, it will contain many subtopic discussions that are never mentioned in its abstract.

When a user engages in a similarity search against a collection of abstracts, the user is in effect specifying that the system find documents whose combination of main topics is most like that of the query. In other words, when abstracts are compared via the vector-space model, they are positioned in a multi-dimensional space where the closer two abstracts are to one another, the more topics they are presumed to have in common. This is often reasonable because when comparing abstracts, the goal is to discover which pairs of documents are most alike. For example, a query against a set of medical abstracts which contains terms for the name of a disease, its symptoms, and possible treatments is best matched against an abstract with as similar a constitution as possible.

Most full text documents are rich in structure. One way to view an expository text is as a sequence of subtopics set against a "backdrop" of one or two main topics. A long text can be comprised of many different subtopics which may be related to one another and to the backdrop in many different ways. The main topics of a text are discussed in its abstract, if one exists, but subtopics usually are not mentioned. Therefore, instead of querying against the entire content of a document, a user should be able to issue a query about a coherent subpart, or subtopic, of a full-length document, and that subtopic should be specifiable with respect to the document's main topic(s).

Figure 3.1 illustrates some of the possible distributional relationships between two terms in the main topic/subtopic framework. An information access system should be aware of each of the possible relationships and make judgments as to relevance based in part on this information. Thus a document with a main topic of "cold fusion" and a subtopic of "funding" would be recognizable even if the two terms do not overlap perfectly. The reverse situation would be recognized as well: documents with a main topic of "funding policies" with subtopics on "cold fusion" should exhibit similar characteristics.

Note that a query for a subtopic in the context of a main topic should be considered to be qualitatively different from a conjunction. A conjunction should specify either a join of two or more main topics or a join of two or more subtopics – it should imply conjoining two like items. In contrast, "in the context of" can be thought of as a subordinating relation (see Figure 3.1).

The idea of the main topic/subtopic dichotomy can be generalized as follows: different distributions of term occurrences have different semantics; that is, they imply different things about the role of the terms in the text.

Consider the chart in Figure 3.2. It shows the possible interlinking of distributions of

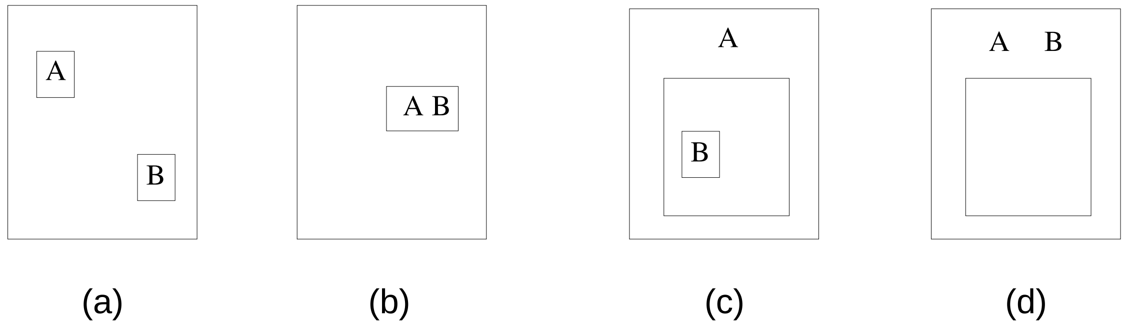


Figure 3.1: Possible relationships between two terms in a full text. (a) The distribution is disjoint, (b) co-occurring locally, (c) term A is discussed globally throughout the text, B is only discussed locally, (d) both A and B are discussed globally throughout the text.

two term sets, Term Set 1 and Term Set 2, where a term set is a set of terms that bear some kind of semantic relationship to one another (e.g., *election*, *poll*, and *vote* or *barney*, *dinosaur*, and *cloying*). The term sets are considered to be symmetric; that is, neither one is more important than the other, and so the lower triangle of the chart is omitted. Within a document, each term set can be characterized as belonging to one of four possible frequency ranges: high, medium, low, and zero, and one of two distribution patterns: global and local. (Term sets with frequency zero are not considered in the chart.) The frequencies are meant to be relative to the length of the document, and the difference between high, medium, and low should be thought of as graded.

For the purposes of interpreting term set distribution it is convenient to assume that the documents have been divided into TextTiles: adjacent, non-overlapping multi-paragraph units of text that are assumed to correspond roughly to the subtopic structure of the text. The distinction between global and local distribution is also meant to be relative to document length. A term set with low frequency and local distribution occurs in one or two tiles; a term set with medium frequency and local distribution occurs in perhaps two groupings of two tiles each, or one grouping of one to three tiles. On the other hand, a term set with medium frequency and global distribution will have terms in roughly half the tiles.

With the aid of this chart we can form hypotheses about the role of interactions among term distribution and frequency and their relationship to document relevance (assuming that if the terms in a term set occur with high frequency then they are globally distributed):

A Instances of both term sets occur with high frequency, or one term set is highly frequent and the other has medium frequency. The document describes both term set concepts to a large extent; this would be useful for a user who wants a main topic discussion of both concepts simultaneously.

B Term set 2 is quite frequent, Term Set 1 infrequent and scattered; probably useful only if the user is primarily interested in Term Set 1.

C Term Set 2 is quite frequent, Term Set 1 infrequent but, as opposed to type B, is

TERM SET 2 \ TERM SET 1	High Global	Medium Global	Medium Local	Low Global	Low Local
High Global	A	A	A	B	C
Medium Global		D	E	F	G
Medium Local			H	I	H
Low Global				J	I
Low Local					H

Figure 3.2: Frequency and distributional relationships between two term sets. See the text for an explanation of the letter labels.

locally organized. There is probably a brief but real discussion of Term Set 1 in relation to Set 2, perhaps a subtopic to 2's main topic. If the frequency is extremely low (e.g, 1), then this is probably a passing reference.

D Both terms are of medium frequency but globally distributed. Most likely the same situation as A, but somewhat less likely to be fully about both term sets.

E Both term sets have medium frequency; one is locally distributed and one globally. If they have some tiles with significant overlap then the document is probably of interest if the user is interested in a main topic/subtopic-like distribution.

F Term Set 2 has medium frequency, Term Set 1 is infrequent, and both are scattered. The two might bear a relationship to one another but there is not enough evidence to decide either way. Less likely to be useful than in G.

G Term Set 2 has medium frequency, globally distributed, and Term Set 1 is infrequent but localized. If the two overlap there is a good chance of a discussion involving both term sets but with only a brief reference to Term Set 1.

H Both term sets have medium or low frequency and are localized. If they overlap then this has some chance of being a good isolated discussion. If they do not overlap, the document should be discarded.

I Both term sets are infrequent, one localized, one not. This document should probably be discarded.

J Both term sets are infrequent and globally distributed. This document should probably be discarded.

Of course these observations should be generalized to more than two term sets, but for multiple term sets the implications of each combination are less clear.

Interestingly, Grimes (1975) had the prescience to suggest the value of localized information as determined by discourse structure. He wrote in 1975:

> Now that information retrieval is taking on greater importance because of the proliferation of circulated information, linguistics may have something to contribute to it through discourse studies. In the first place, studies of discourse seem to show that the essential information in some discourses is localized, which implied that for retrieval it might be possible to specify parts of the discourse that do not have to be taken into account. There is definitely a pattern of organization of information in any discourse that can be recognized and should therefore be explored for its usefulness in retrieval; for example, Halliday's notion of the distribution of given and new information.

Grimes' suggestion of using the localized structure of discourse to eliminate certain passages is a useful one, although different than that suggested here. Work along related lines does appear in Liddy (1991), which discusses the usefulness of understanding the structure of an abstract when using a natural-language based information retrieval approach, and Liddy & Myaeng (1993), which uses information about the kind of sentence a term occurs in; e.g., differentiating terms that occur in background sentences from those that occur in spoken quotations and those that are in lead sentences in order to better understand the relations among terms.

Given the analysis surrounding the chart of Figure 3.2, how can these observations about relative term distribution be incorporated into an information access system? The following section discusses this issue, first touching on problems with existing approaches, and then suggesting a new solution.

3.4 Distribution-Sensitive Information Access

3.4.1 The Problem with Ranking

Noreault *et al.* (1981) performed an experiment on bibliographic records in which they tried every combination of 37 weighting formulas working in conjunction with 64 combining formulas on Boolean queries. They found that the choice of scheme made almost no difference: the best combinations got about 20% better than random ordering, and no one scheme stood out above the rest.

These results imply that small changes to weighting formulas don't have much of an effect. As found in other aspects of text analysis for information retrieval, (e.g., effects of stemming, or morphological analysis, or using phrases instead of isolated terms), a modification of an algorithm improves the results in some situations and degrades the results in others.

Why might this be the case? Perhaps the answer is that *there is no single correct answer*. Perhaps trying to assign numbers to the impoverished information that we have about the documents (or in this case of the experiment in Noreault *et al.* (1981), bibliographic records) is not an appropriate thing to do. It could be the case that when different kinds of information are present in the texts the term ranking serves only to hide this information from the user. Rather than hiding what is going on behind a ranking strategy, I contend it is better to show the users what has happened as a result of their query and allow the users to determine for themselves what looks interesting or relevant. Of course, this is the intended goal of ranking. But an ordered list of titles and probabilities is under-informative. The link between the query terms, the similarity comparison, and the contents of the texts in the dataset is too underspecified to assume that a single indicator of relevance can be assigned.

Instead, the representation of the results of the retrieval should present as many attributes of the texts and their relationship to the queries as possible, and present the information in a compact, coherent and accurate manner. Accurate in this case means a true reflection of the relationship between the query and the documents.

Consider for example what happens when one performs a keyword search using WAIS (Kahle & Medlar 1991). If the search completes, it results in a list of document titles and relevance rankings. The rankings are based on the query terms in some capacity, but it is unclear what role the terms play or what the reasons behind the rankings are. The length of the document is indicated by a number, which although interpretable, is not easily read from the display. Figure 3.3 represents the results of a search on *image* and *network* on a database of conference announcements. The user cannot determine to what extent either term is discussed in the document or what role the terms play with respect to one another. If the user prefers a dense discussion of images and would be happy with only a tangental reference to networking, there is no way to express this preference.

Attempts to place this kind of expressiveness into keyword based system are usually flawed in that the users find it difficult to guess how to weight the terms. If the guess is off by a little they may miss documents that might be relevant, especially because the role the weights play in the computation is far from transparent. Furthermore, the user may be willing to look at documents that are not extremely focused on one term, so long as the references to the other terms are more than passing ones. Finally, the specification of such information is complicated and time-consuming.

The concern in the information retrieval literature about how to rank the results of Boolean and vector space-type queries is misplaced. Once there is a baseline of evidence for choosing a subset of the thousands of available documents, then the issue becomes a matter of providing the user with information that is informative and compact enough to be able to be interpreted swiftly. As discussed in the previous section, there are many different ways a long text can be "similar" to the query that issued it, and so we need to supply the user with a way to understand the relationship between the retrieved documents and the query.

3.4.2 Analogy to Problems with Query Specification

There have been many studies showing that users have difficulty with Boolean logic queries and many attempts at making the query formulation process easier. Research papers discuss at great length the relative benefits of one query language over another. However, this issue is circumvented to some extent if instead a system provides the user with an intuitive, direct-manipulation interface (Shneiderman 1987).

A good example of this is the difference between the keyword-based interface to large online bibliographic systems. The user has to remember the correct keywords to use from system to system, and must remember where to place AND and OR connectives. For example, with MELVYL, the online bibliographic system for the University of California (Lynch 1992), to look for a book by de Tocqueville containing the word *Democracy*, one must enter

```
fi pa tocqueville and tw democracy
```

where *pa* indicates "personal author" and *tw* indicates "title words". However, to find a title with both words *democracy* and *america* one need enter only one copy of the keyword *tw*:

```
fi tw democracy america
```

```
image network

This is a searchable index.  Enter search keywords:

Index conf.announce contains the following 164 items relevant to
'image network'.  The first figure for each entry is its relative
score, the second the number of lines in the item.

* 1000 1190 /ftp/pub/conf.announce/jenc5
* 886 125 /ftp/pub/conf.announce/image.processing.conf
* 800 334 /ftp/pub/conf.announce/image.analysis.symposium
* 743 303 /ftp/pub/conf.announce/sans-III
* 543 376 /ftp/pub/conf.announce/atnac.94
* 486 133 /ftp/pub/conf.announce/sid
* 486 125 /ftp/pub/conf.announce/qes2
* 457 138 /ftp/pub/conf.announce/europen.forum.94
* 429 378 /ftp/pub/conf.announce/mva.94
* 429 785 /ftp/pub/conf.announce/openview.conf
* 429 104 /ftp/pub/conf.announce/high.performance.networking
* 400 217 /ftp/pub/conf.announce/nonlinear.signal.workshop
* 429 378 /ftp/pub/conf.announce/vision.interface.94
* 429 785 /ftp/pub/conf.announce/inet.94
* 429 104 /ftp/pub/conf.announce/icmcs.94
* 400 217 /ftp/pub/conf.announce/internetworking.94
* 371 220 /ftp/pub/conf.announce/iss.95
* 371 168 /ftp/pub/conf.announce/qes1
* 343 152 /ftp/pub/conf.announce/conti.94
* 343 247 /ftp/pub/conf.announce/elvira
```

Figure 3.3: A sketch of the results of a WAIS search on *image* and *network* on a dataset of conference announcements.

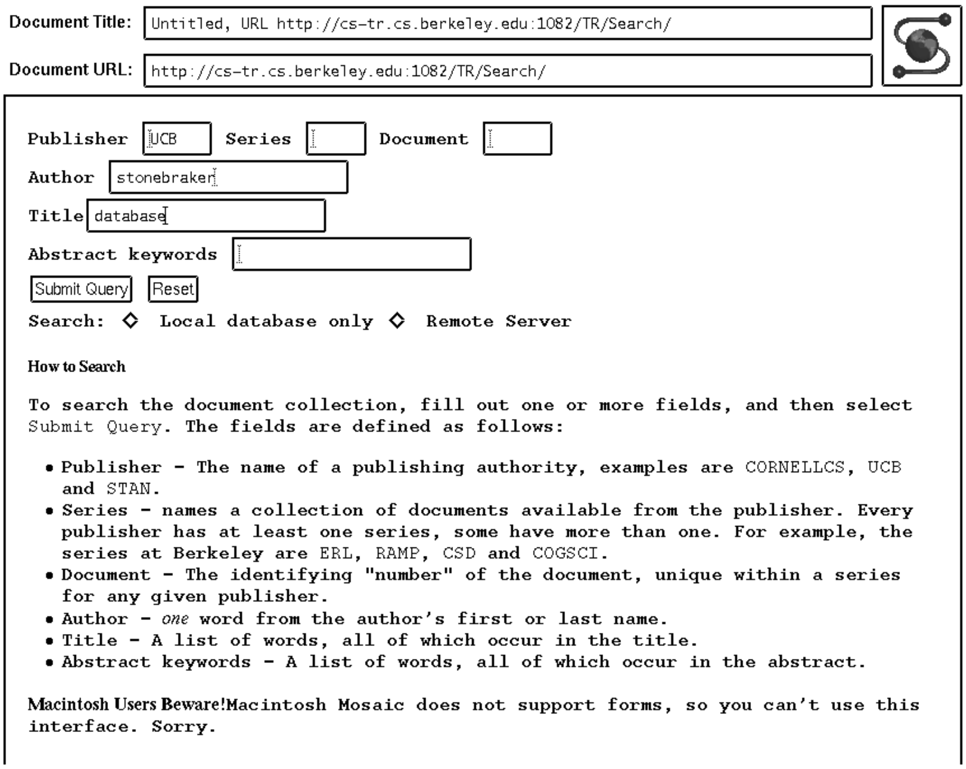
(although it is legal to enter "fi tw democracy and tw america" this results in a much longer search due to the indexing structure underlying the system (Farley 1989)). However, to find a book by two authors, two copies of the keyword *pa* (and the AND connective) must be used, as demonstrated by the error below:

```
CAT-> find pa mosteller wallace

Search request: FIND PA MOSTELLER WALLACE
Search result:  0 records at all libraries

Please type HELP

CAT-> find pa mosteller and pa wallace

Search request: FIND PA MOSTELLER AND PA WALLACE
Search result:  2 records at all libraries

Type D to display results, or type HELP.
```

GLADIS is the other major bibliographic system available at UC Berkeley, indexing mainly the local collection and providing timely information such as check-out status. Unfortunately, its keyword list is slightly different and the interface is unforgiving with respect to this:

```
===> find pa tocqueville
**>   THE SEARCH CODE WAS NOT RECOGNIZED
**>   Type a search code listed above
**>
===> find pn tocqueville


Your search for the Personal Name: TOCQUEVILLE
retrieved 41 name entries.
```

Another problem with these systems is that although they have some very powerful special purpose search capabilities (such as the capability to look for PhD dissertations specifically, in the case of MELVYL) users are unaware of the options because they require knowledge of special keywords.



Document Title: Untitled, URL http://cs-tr.cs.berkeley.edu:1082/TR/Search/

Document URL: http://cs-tr.cs.berkeley.edu:1082/TR/Search/

Publisher UCB Series Document

Author stonebraker

Title database

Abstract keywords

Submit Query Reset

Search: ◇ Local database only ◇ Remote Server

How to Search

To search the document collection, fill out one or more fields, and then select Submit Query. The fields are defined as follows:

- Publisher - The name of a publishing authority, examples are CORNELLCS, UCB and STAN.
- Series - names a collection of documents available from the publisher. Every publisher has at least one series, some have more than one. For example, the series at Berkeley are ERL, RAMP, CSD and COGSCI.
- Document - The identifying "number" of the document, unique within a series for any given publisher.
- Author - *one* word from the author's first or last name.
- Title - A list of words, all of which occur in the title.
- Abstract keywords - A list of words, all of which occur in the abstract.

Macintosh Users Beware!Macintosh Mosaic does not support forms, so you can't use this interface. Sorry.

Figure 3.4: A fill-in-the-forms type interface to a bibliographic dataset.

Most query-formulation problems can be circumvented via a forms-based interface. Davis (1994) has recently developed such an interface to the bibliographic records of the CNRI computer science online technical report project (see Figure 3.4). In this interface, the options are spelled out clearly, all options are visible, and the interface itself supplies the syntax for the query. A similar situation arises in the world of database management systems. Much effort has been expended on trying to determine the right way to formulate keyword-and-syntax based query languages, when evidence suggests that graphically-oriented ways of specifying the query are preferable for most kinds of queries (Bell & Rowe 1990).

There is an analogy between systems that require obscure keyword languages and systems that display results based on an invisible ranking algorithm. Neither supply the user with a representation that reflects the underlying information. Both probably arose due to the limitations of computer hardware at the time, and unfortunately are still in use today.

3.4.3 TileBars

This section presents one solution to the problems described in the previous subsections. The approach is synthesized in reaction to three hypotheses discussed earlier:

- Long texts differ from abstracts and short texts in that, along with term frequency, term distribution information is important for determining relevance.
- The relationship between the retrieved documents and the terms of the query should be presented to the user in a compact, coherent, and accurate manner (as opposed to the single-point of information provided by a ranking).
- Passage-based retrieval should be set up to provide the user with the context in which the passage was retrieved, both within the document, and with respect to the query (this issue is discussed in more detail in Section 3.5).

Figure 3.5 shows an example of a new representational paradigm, called *TileBars*, which provides a compact and informative iconic representation of the documents' contents with respect to the query terms. TileBars allow users to make informed decisions about not only which documents to view, but also which passages of those documents, based on the distributional behavior of the query terms in the documents. The goal is to simultaneously indicate the relative length of the document, the relative frequency of the term sets in the document, their distribution with respect to the document, and their distribution with respect to each other. Each large rectangle indicates a document, and each square within the document represents a TextTile. The darker the tile, the more frequent the term (white indicates 0, black indicates 9 or more instances, the frequencies of all the terms within a term set are added together). Since the bars for each set of query terms are lined up one next to the other, this produces a representation that simultaneously and compactly indicates relative document length, query term frequency, and query term distribution. The representation exploits the natural pattern-recognition capabilities of the human perceptual system (Mackinlay 1986); the patterns in a column of TileBars can be quickly scanned and deciphered. I hypothesize that the interpretation of the patterns should be along the lines outlined in Section 3.3. Some case studies appear in Section 3.4.4 below.

Term overlap and term distribution are easy to compute and can be displayed in a manner in which both attributes together create easily recognized patterns. For example, overall darkness indicates a text in which both term sets are discussed in detail. When both term sets are discussed simultaneously, their corresponding tiles blend together to cause a prominent block to appear. Scattered discussions have lightly colored tiles and large areas of white space. Note that the patterns that can be seen here bear some resemblence to Figure 2.4 in Chapter 2, in which term distributions for a text are displayed.

TileBars make use of the following visualization properties (extracted from Senay & Ignatius (1990)):

- A variation in position, size, value [gray scale saturation], or texture is ordered [ordinal] that is, it imposes an order which is universal and immediately perceptible. (Bertin 1983)
- A variation in position, size, value [gray scale saturation], texture or color is selective, that is, it enables us to isolate all marks belonging to the same category. (Bertin 1983)

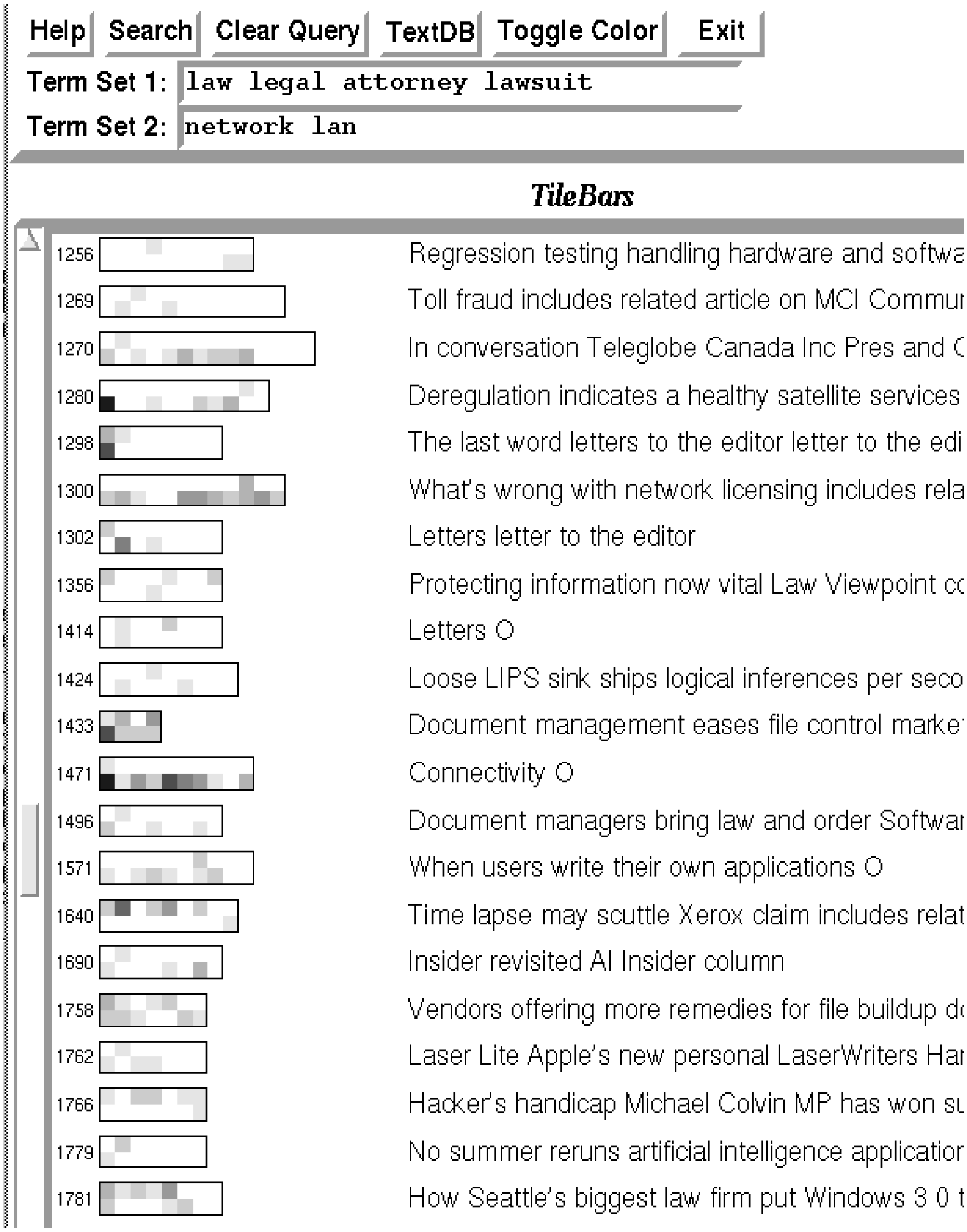


Figure 3.5: The TileBar display paradigm. Rectangles correspond to documents, squares correspond to TextTiles, the darkness of a square indicates the frequency of terms in the corresponding Term Set. Titles and the initial words of a document appear next to its TileBar. Term Set 1 consists of *law*, *legal*, *attorney*, *lawsuit* and Term Set 2 consists of *network* and *lan*.

- If shading is used, make sure differences in shading line up with the values being represented. The lightest ("unfilled") regions represent "less", and darkest ("most filled") regions represent "more". (Kosslyn *et al.* 1983)
- Because they do have a natural visual hierarchy, varying shades of gray show varying quantities better than color. (Tufte 1983)

Note that the stacking of the terms in the query-entering portion of the document is reflected in the stacking of the tiling information in the TileBar: the top row indicates the frequencies of terms from Term Set 1 and the bottom row corresponds to Term Set 2. Thus the issue of how to specify the keyterms becomes a matter of what information to request in the interface.

TileBars allow the user to be aware of what part of the document they are about to view before they view it. If they feel they need to know more of what the document is about they can simply mouse-click on a part of the representation that symbolizes the beginning of the document. If they wish to go directly to a tile in which term overlap occurs, they click on that portion of the text, knowing in advance how far down in the document the passage occurs.

The issue of how to rank the documents, if ranking is desired, becomes clearer now. Documents can be grouped by distribution pattern, if this is found to be useful for the user. Each pattern type can occupy its own window in the display and users can indicate preferences by virtue of which windows they use. Thus there is no single correct ranking strategy: in some cases the user might want documents in which the terms overlap throughout; in other cases isolated passages might be appropriate. Figure 3.7 shows an example in which a query's retrieval results have been organized by distribution pattern type.

Relevance feedback is generally perceived as an effective strategy for improving the results of retrieval (Salton & Buckley 1990). In relevance feedback, the system responds to input from the user indicating which documents are of interest and which are to be discarded. From this information the system can guess how to downweight some terms and increase the weight on other terms, as well as introduce new terms into the query based on the documents that the user found especially helpful. Relevance feedback appears to work well because the user helps set term weights, indirectly specifying which formulas better describe the kind of information being sought. However, the gathering of relevance feedback is time-consuming and draining on the user, since it requires the user to read the text for content and guess whether or not the terms of the document will be useful for finding other interesting documents.

TileBars could provide a relevance feedback mechanism in which users can indicate patterns of interest as well as or instead of terms of interest. Relevance feedback based on patterns should be more effective than requiring a specification of what kinds of patterns are desired in advance, or requiring the entry of a query in terms of subtopic/main topic or some other relationship. It could also act as an alternative or a supplement to relevance feedback on term similarity, since as argued above, overall similarity is less likely to be useful for long texts, with their varied internal structure, than abstracts. However, this idea has not yet been implemented.

TileBars display context corresponding directly to the users' query; specifically to the terms used in a free-text search. Sometimes, however, the user is unsure of what kind of queries to make and needs to get familiar with new textbases rapidly. Chapter 5 describes the use of main topic information to help provide context in this situation.

Implementation Notes

The current implementation of the information access method underlying the TileBar display makes use of $\approx$ 3800 texts of length 100-500 lines from the ZIFF portion of the TIPSTER corpus and $\approx$ 250 texts of the same length from the AP portion of TIPSTER, for a total of about 57Mbytes (Harman 1993). (ZIFF is comprised mainly of commercial computer news and AP is world news from the late 1980s.) The interface was written using the Tcl/TK X11-based toolkit (Ousterhout 1991). The search engine makes use of customized inverted index code created especially for this task[2]; each term is indexed by document and tile number, and the associated frequencies. In the future this may be replaced with the POSTGRES database management system, which has support for large objects and user-defined types (Stonebraker & Kemnitz 1991). An alternative indexing stratum is that of GLIMPSE (Manber & Wu 1994) (built on agrep Wu & Manber (1992)) which stores a small index (about 2-4% of the size of the text collection) but has an acceptable speed for many tasks.

The informativeness of the TileBar representation is hindered when the results of tiling are inaccurate. The ZIFF database contains many documents comprised of lists of concatenated short news articles, and some documents comprised of single-line calendar items. The tiling algorithm is set up so that a single line segment is too fine a division; therefore, documents like the calendar text will have erroneous tilings (although arbitrary groupings on terms like these are perhaps preferable to assigning each sentence its own tile, due to efficiency considerations). The algorithm does do fairly well at distinguishing slightly longer concatenated articles, such as sequences of paragraph-long news summaries and letters to the editor. It is also quite good at recognizing the boundaries of summarizing information at the beginning of articles when such information appears.

3.4.4 Case Studies

This section examines the properies of TileBars in more detail, using two example queries on the ZIFF corpus.

---

[2] I am grateful to Marc Teitelbaum for the swift implementation of this code.

Networks and the Law

Figure 3.5 shows some of the TileBars produced for the query on the term sets *law legal attorney lawsuit* and *network lan*. In this portion of the ZIFF collection, the terms of interest have the following averages of occurrence, in the documents in which they appear at least once:

	$\bar{s}$	σ
legal	2.4	3.6
law	2.8	4.2
attorney	1.5	1.0
lawsuit	2.3	3.5
network	10.7	5.2
lan	6.8	10.2

What kind of documents can we expect to find in response to this query? Use of computer networks by law firms, lawsuits involving illegal use of networks, and patent battles among network vendors are all possibilities that come to mind. We know that since we are searching in a collection of commercial computer documents, most instances of the word *network* will refer to the computer network sense, with exceptions for telephone systems, neural networks, and perhaps some references to the construct used in theoretical analyses. Since *legal* is an adjective, it can be used as a modifier in a variety of situations, but together with the other terms in its set, a large showing of these terms should indicate a legitimate instance of a discussion in the legal frame. These two term sets were specifically chosen because their meanings are usually in quite separate semantic frames; the next example will discuss a query involving terms that are more related in meaning.

In Figure 3.5, the results have not been sorted in any manner other than document ID number. It is instructive to examine what the bars indicate about the content of the texts and compare that against the hypothesis of Section 3.3 and against what actually is discussed in the texts. Document 1433 jumps out because it appears to discuss both term sets in some detail (type A from the chart). Documents 1300 and 1471 are also prominent because of a strong showing of the network term set (type C). Document 1758 also has well-distributed instances of both term sets, although with less frequency than in document 1433 (type H). Legal terms have a strong distributional showing in 1640, 1766, 1781 as well (types C and G). We also note a large number of documents with very few occurrences of either term, although in some cases terms are more locally concentrated than in others. Document 1298 is interesting in that it seems to have an isolated but intense discussion of both term sets (type H); the fact that neither term set continues on into the rest of the document implies that this discussion is isolated from the rest in meaning as well. Most of the other documents look uninteresting due to their lack of overlap or infrequency of term occurrences.

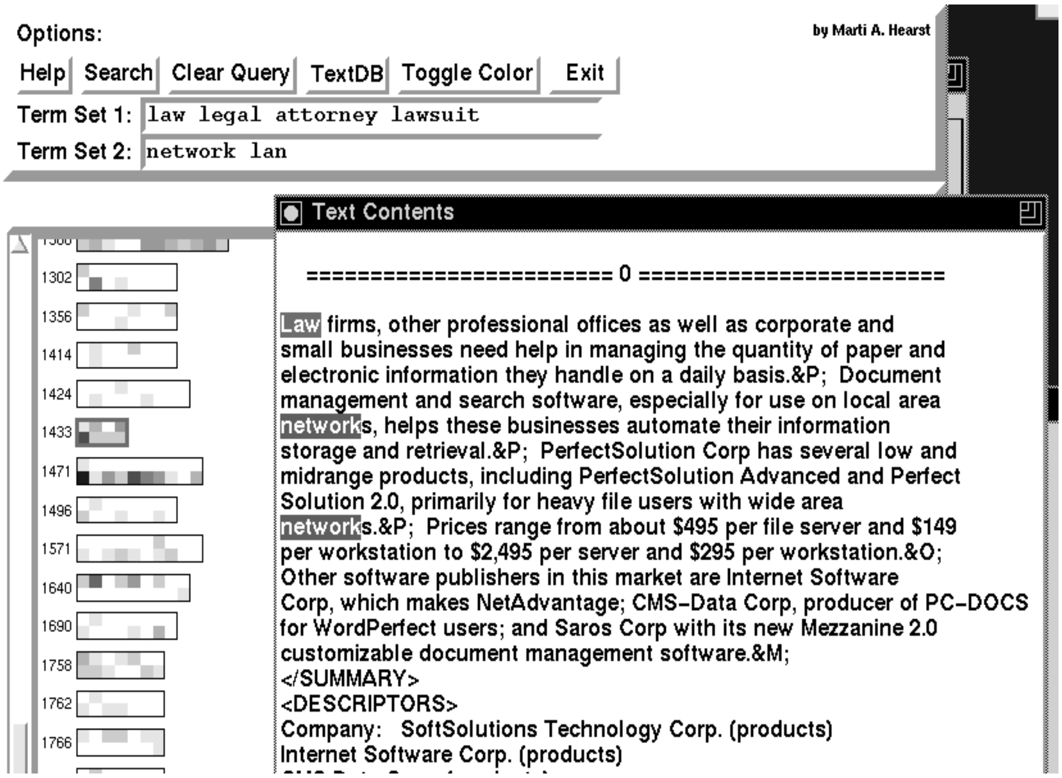


Figure 3.6: The results of clicking on the first tile of document 1433: the search terms are highlighted and the tile number is shown.

Looking now at the actual documents we can determine the accuracy of the inferences drawn from the TileBars. Clicking on the first tile of document 1433 brings up a window containing the contents of the document, centered on the first tile (see Figure 3.6). The search terms are highlighted with two different colors, distinguished by term set membership, and the tile boundaries are indicated by ruled lines and tile numbers. The document describes in detail the use of a network within a legal office.

Looking at document 1300, the intersection between the term sets can be viewed directly by clicking on the appropriate tile. From the TileBar we know in advance that the tile to be shown appears about three quarters of the way through the document. Clicking here reveals a discussion of legal ramifications of licensing software when distributing it over the network.

Document 1471 has only the barest instance of legal terms and so it is not expected to contain a discussion of interest – most likely a passing reference to an application. Indeed, the term is used as part of a hypothetical question in an advice column describing how to configure LANs.

The expectation for 1758 is that it will discuss both term sets, although not as intensely as did 1433. Since some of the term instances concentrate near the beginning of this document, selecting this viewing point seems sensible, yielding a discussion of a documentation management system on a networked PC system in a legal office.

The remaining documents with strong distributions of legal terms – IDs 1640, 1766, 1781 – discuss a lawsuit between software providers, computer crime, and another discussion of a law firm using a new networked software system, respectively. Appropriately, only the latter has overlap with networking terms, since the other two documents do not discuss networking in the legal context. Interestingly, the solitary mention of networking at the end of 1766 lists it as a computer crime problem to be worried about in the near future. This is an example of the suggestive nature of the positional information inherent in the representation.

Finally, looking at the seemingly isolated discussion of document 1298 we see a letter-to-the-editor about the lack of liability and property law in the area of computer networking. This letter is one of several letters-to-the-editor; hence its isolated nature. This is an example of a perhaps useful instance of isolated, but strongly overlapping, term occurrences. In this example, one might wonder why one legal term continues on into the next tile. This is a case in which the tiling algorithm is slightly off in the boundary determination.

As mentioned above, the remaining documents appear uninteresting since there is little overlap among the terms and within each tile the terms occur only once or twice. We can confirm this suspicion with a couple of examples. Document 1270 (type F/G) has one instance of a legal term; it is a passing reference to the former profession of an interview subject. Document 1356 (type I/H) discusses a court's legal decision about intellectual property rights on information. Tile 3 provides a list of ways to protect confidential information, one item of which is to avoid storing confidential information on a LAN. So in this case the reference is relevant if not compelling.

Figure 3.7 shows the results of the same query when placed in an interface that sorts the terms according to their frequency and patterns of distribution. The upper lefthand window displays the documents in which both term sets occur in at least 40% of the tiles. The upper righthand window shows those documents in which at least 40% of the tiles have occurrences of terms from Term Set 1, but occurrences from Term Set 2 are less well-distributed. The lower lefthand window shows the symmetric case, and the lower righthand window displays the documents in which neither term occurs in more than 40% of the tiles. Within each window the documents are sorted by overall query term frequency. Experiments need to be run to evaluate the effectiveness of variations in pattern criteria.

CD-ROMs and Games

Section 3.3 hypothesized about the role of medium frequency terms. This example examines how term distribution can make a difference in whether or not two term sets stand in a modificational relationship. In response to a query on *cd-rom* and *game*, 49 documents were retrieved. Figure 3.8 shows a clip of some of the documents' TileBars.

Viewed by frequency alone, documents 2238 and 2003 seem equally viable (or not viable):

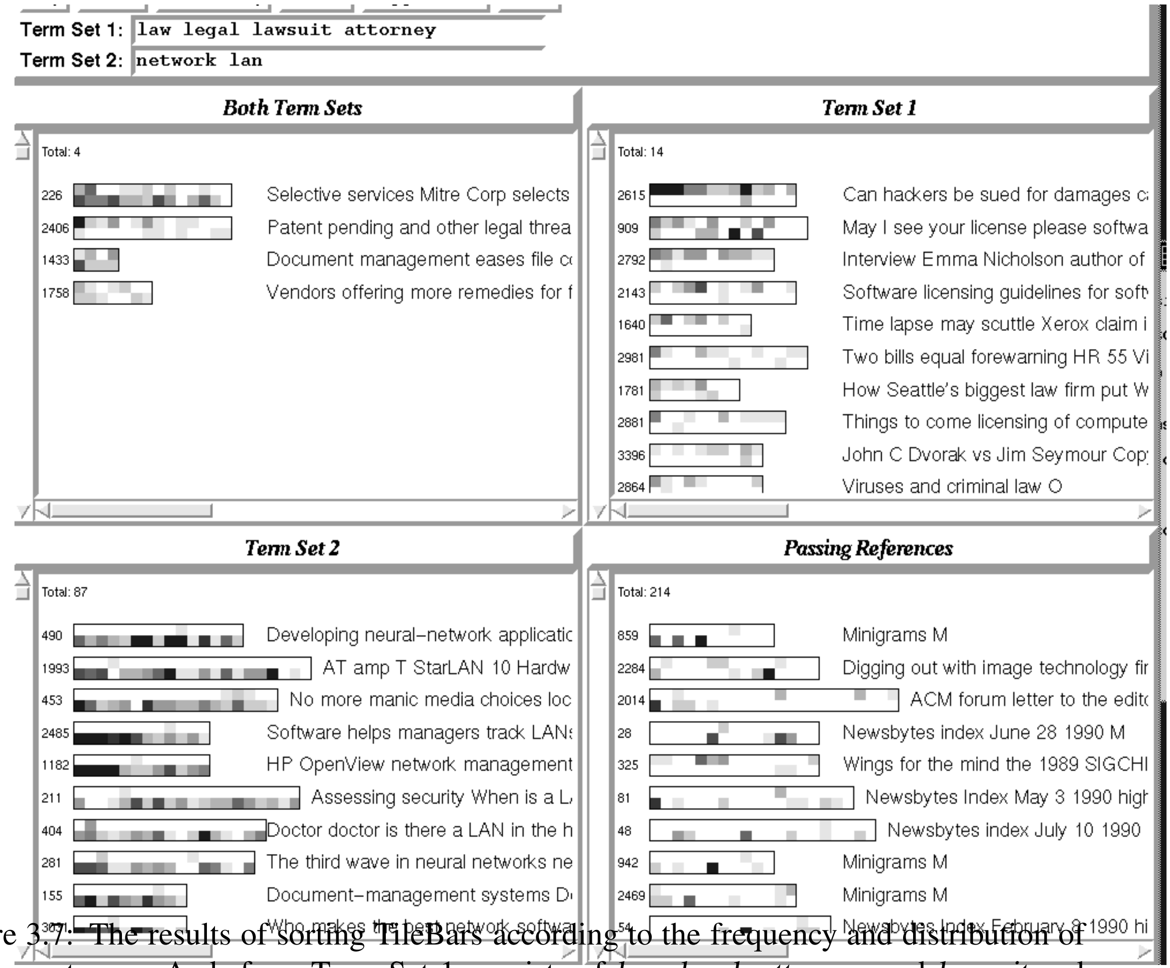


Figure 3.7: The results of sorting TileBars according to the frequency and distribution of the query terms. As before, Term Set 1 consists of *law*, *legal*, *attorney*, and *lawsuit* and Term Set 2 consists of *network* and *lan*.

Doc: 2238 cd-rom: 13 game: 2

Doc: 2003 cd-rom: 9 game: 3

However, taking into account the number of tiles each term occurs in changes the picture:

Doc: 2238 cd-rom: 13 10/25 tiles game: 2 2/25 tiles

Doc: 2003 cd-rom: 9 2/20 tiles game: 3 3/20 tiles

We see that the references to *cd-rom* in 2238 are quite spread out, whereas those in 2003 are quite localized. The only question that remains is whether or not the localized discussions of *cd-rom* in 2003 coincide with those of *game*. From the context bar we can easily see that they do not, and so we assume the document is not of interest. The discussion in 2238 might also be bunched together, as is the case in 1808, but in this case it is more spread out and we can guess that the use of *game* in this context bears at least some meaningful relationship to CD-ROMs.

Upon inspecting the documents, we see that 2003 consists of a sequence of disjoint newsbites, whereas 2238 describes applications of CD-ROM technology, including a golf game application. Also verifying our suspicions about document 1808, we see that the lagging use of *game* here, far away from all the *cd-rom* references, is a metaphorical one about predicting prices for WORM devices ("a dart-throwing game"). Note, however, that there would have been some overlap in this case if the query had been on *worm* and *game*, but it will again have appeared to be a passing reference.

This diagram has another interesting case in which it seems clear that a dense discussion of the two terms takes place, although for only part of the document, in document 3753. Clicking in the middle of this discussion indeed reveals a discussion of the use of CD-ROMs for game play.

The first tile of document 1669 leads into a discussion of the utility of CD-ROM technology by mentioning a list of applications, including games, an encyclopedia, and music-appreciation software. And not surprisingly, due to the pattern of intensities of the term occurrences, document 3811 is a review of varies CD-ROM-based games.

From these examples it should seem likely that with very little exposure a user can become fluent in interpreting TileBars.

3.5 Passage-based Information Access

This chapter has alluded to issues relating to passage-level information access; this section discusses some general issues and the more conventional approaches to passage retrieval. To date there has been little research on passage retrieval, most likely for the reasons stated at the beginning of the chapter; especially the lack of available online full text for experimentation. An accompanying important fact is that there are no passage

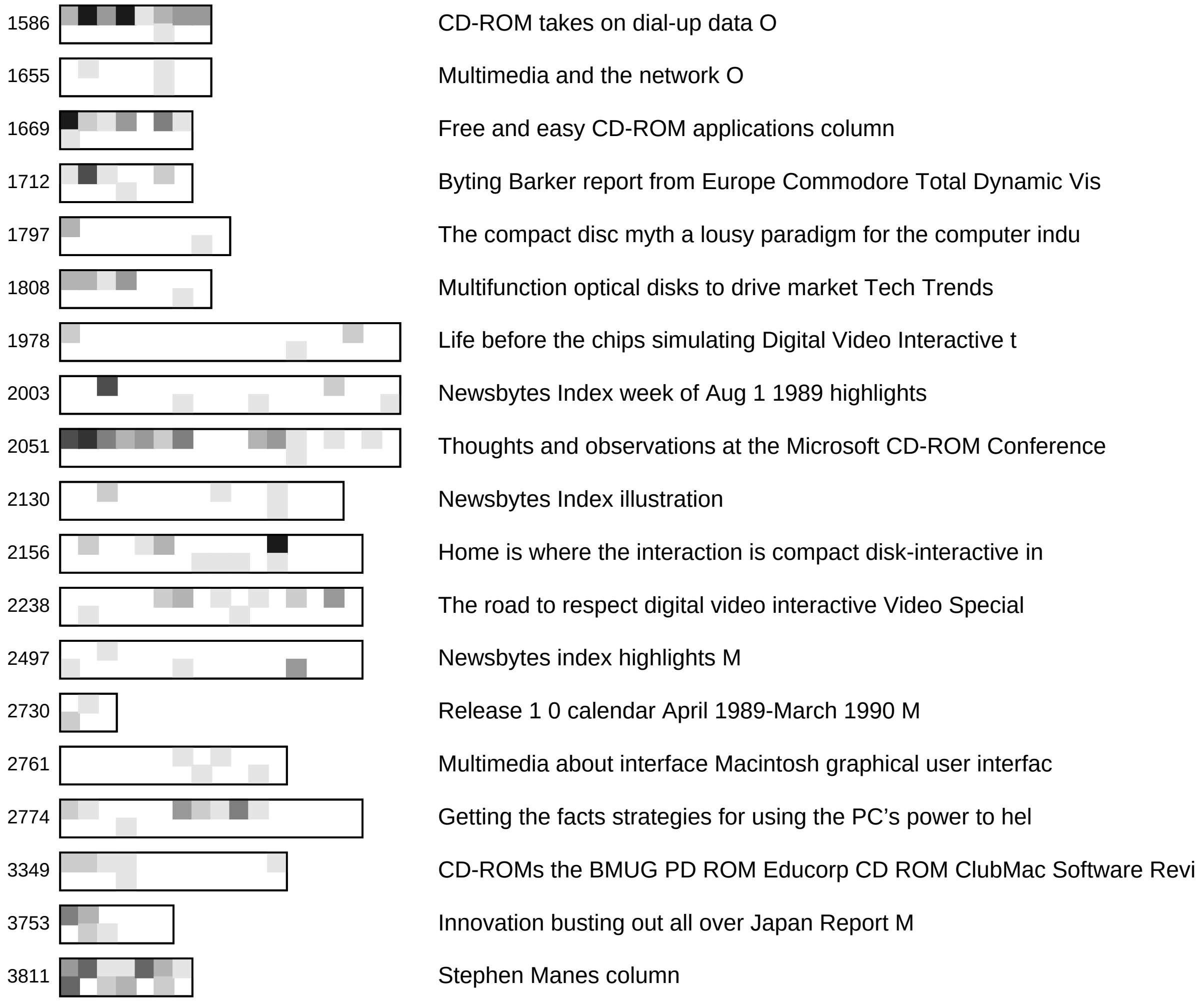


Figure 3.8: Some TileBars found in response to a query in which Term Set 1 is *cd-rom* and Term Set 2 is *game*.

retrieval test sets; that is, no test sets in which portions of long texts have been identified as relevant for a query set. The closest available is the recent TIPSTER/TREC collection and relevance judgments (Harman 1993), but although this collection includes some long documents, it does not include relevance judgments for passages alone.

Several questions need to be addressed in the study of passage retrieval, related to the discussions of the previous sections. For example, given a retrieved passage, where in the text did the passage come from: the beginning, middle, end of the document? How are a passage's neighbors in a document related to it? Was the passage chosen because of its contribution in isolation to the relevance of the document or is it just a representative part, and if so, representative in what way? If chosen for a Boolean query, how much and in what context does each term of the query contribute? There is a need for a test collection for passage retrieval that is sensitive to these kinds of distinctions.

Researchers working with hypertext have explored issues pertaining to organizing information within one or a few long documents, but have not focused on issues related to presenting isolated pieces of texts drawn from a large collection of texts. Fuller *et al.* (1993), in discussing strategies for hypertext, make the important suggestion of providing context for the text nodes that are retrieved as a result of a query, rather than just presenting a list of relevant nodes. They contrast the approach in standard information retrieval, in which the structure is not accessible to the similarity engine or viewable by the users, with hypertext systems that do not provide good search capabilities or sophisticated storage systems. They do not supply viable solutions to the problem, however.

3.5.1 An Analysis of two TREC Topic Descriptions

As mentioned above, the relevance judges for TREC were not concerned with distinguishing retrieval of passages versus retrieval of documents overall. Bearing in mind that only a small percentage of the TREC documents are long, this is not surprising. But the fact that relevance judgments do not refer to particular parts of long documents is problematic for the purposes of training and evaluating passage retrieval algorithms. Another problem with the collection is that the documents have not been ranked according to their relative relevance, so there is no way to know what variations in ranking are to be preferred for a query that has many positive relevance assignments.

It is an illuminating exercise to convert TREC topic descriptions to representations applicable to TileBars. Some of the topic descriptions, although long and detailed, can be addressed by simply finding the documents with a few key terms. For example, all and only the documents in the ZIFF subset that contain the word *superconductivity* are relevant to Topic 021. Many of the topic descriptions require a particular product or company name to be identified, or a company name in conjunction with some other specifically named item. Relevant documents for this kind of topic description often have all the key terms in a single sentence. In these cases only very local parts of a long text need to match in order to satisfy the query. In other cases, topic descriptions require the topics to be discussed throughout the document.

Still other topic descriptions require the mention of a company name or a country or some other proper noun in conjunction with a general topic, e.g., companies working on multimedia systems. This is most likely meant to simulate a filtering or message-stuffing task, as in the MUC competitions (Sundheim 1990). It also requires recognition of country, company, and other proper names. This is a case where distributional information will play a role in some cases, but again often the relevant terms need be found only locally. Still other topics include a context or environment in which a topic is to be discussed has been specified. This kind of topic might benefit from an understanding of term distribution information.

Below I show two examples of TREC queries, their tranformations into TileBar representations, and the different characteristics that can be discerned about the relevant documents using this representation.

Consider the following TREC topic description:

> Topic 005 <dom> Domain: International Economics
>
> <title> Topic: Dumping Charges
>
> <desc> Description:
>
> The U.S. or the EC charges Japan with dumping a product on any market and/or takes action against Japan for proven or suspected dumping.
>
> <narr> Narrative:
>
> To be relevant, a document must discuss current charges made by the U.S. or the EC against Japan for dumping a product on the U.S., EC, or any third-country market, and/or action(s) taken by the U.S. or the EC against Japan for proven or suspected dumping. The product must be identified.
>
> <con> Concept(s):
>
> 1. dumping
>
> 2. duties, tariffs, anti-dumping measures, punitive trade sanctions, protective penalties
>
> 3. below market, unfair, predatory pricing
>
> 4. Commerce Department, International Trade Commission (ITC), European Community (EC), Common Market
>
> 5. ruling, charges, investigation

Figure 3.9 shows the results of searching on *dump dumping anti-dumping* and *japan japanese* in the subset of ZIFF used. The relevance judgments assigned by the TREC judges state that of the visible documents, the following ones are relevant: 1700, 1765, 2184, and 3670. For example, from Document 2184 (ZF07-376-802), which is judged relevant, comes the following passage:

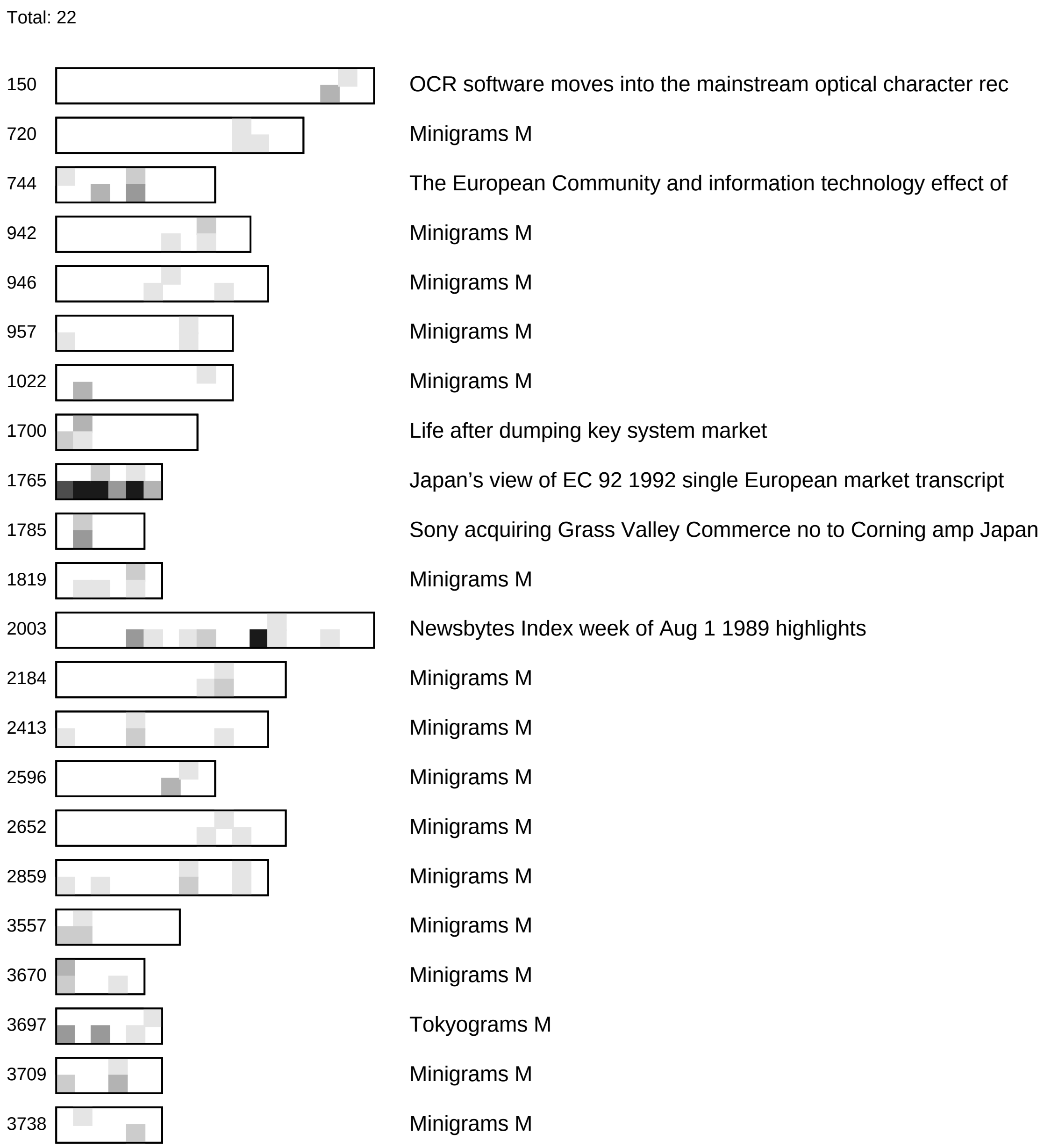


Figure 3.9: TileBars found in response to a simplified version of TREC topic description 005. Term Set 1 = *dump dumping anti-dumping* and Term Set 2 = *japan japanese*.

> [...]
>
> One of the more worrying prospects for 1989 is for a big surge of protectionism, and in the US, AT&T Co has asked the International Trade Commission to look into alleged dumping by manufacturers in *Japan*, South Korea and Taiwan of small PABXs and key systems: AT&T claims that US firms have been severely injured by the practices of more than a dozen Far East manufacturers marketing systems at unfair prices under more than 17 brand names; the companies named in the complaint are Toshiba, Matsushita, Hasegawa, Iwatsu, Meisei, Makayo, Nitsuko and Tamura, all of *Japan*; Goldstar, Samsung and OPC of South Korea, and Sun Moon Star of Taiwan; AT&T says the practices have enabled the companies to raise their share of the market to 60% from 40% since 1985.
>
> [...]

The TileBars for each of these documents display appropriate overlap. But what about the documents whose TileBars indicate overlap, but are not marked relevant? Some of these are documents 2413, 2859, 3557, and 3709. In only one case (2413) does either term set occur frequently, so the others might be irrelevant references. The pertinent fragments are shown below; three out of four could be considered relevant.

In Document 2413 (ZF07-387-928), tile 4:, we find:

> [...]
>
> *Japan* has removed all the controls on exports of memory chips to the European Community in compliance with international trade rules, the European Commission said: the restrictions arose from the controversial third country fair market value provisions of the US-*Japan* Semiconductor Trade Agreement, which were declared illegal under the General Agreement on Tariffs & Trade - but the Commission is still studying possible *dump*ing of memory chips in Europe by *Japan*.
>
> [...]

In Document 2859 (ZF07-755-876), has the following passage:

> [...]
>
> *Japan*ese printer manufacturers Star Micronics and NEC Corp, presently using their UK plants to penetrate the European market, have agreed to increase the number of European components in their machines, so avoiding the European Community anti-*dump*ing taxes recently imposed on them: last week, a sitting of the European Commission found that fewer than 40% of the components came from European firms, and as such the printers came under the same tax ruling as direct imports from *Japan* - around $15 dollars a printer for Star and $33 for NEC; accordingly, both firms have undertaken to include more European components, and if this is accepted at the Commission's next sitting, the taxes will be duly annulled.
>
> [...]

These two passages both seem relevant to the topic description.

In Document 3557 (ZF07-376-770), the tiling is incorrect, because it consists of a series of very short news clips. Perhaps in part for this reason, the document is not relevant:

> [...]
>
> Easing the trade tension a little, the European Commission has
>
> ======================= 1 =======================
>
> lifted the anti-*dump*ing duties on photocopiers assembled with the Community by Toshiba Corp and Matsushita Electronic Industrial Co on the grounds that European content now exceeds 40%: the only company still suffering duties is now Konica Inc.
>
> - o -
>
> For Thorn Ericsson Telecommunications Ltd, read Ericsson Ltd: the
>
> Horsham, Sussex-based company, now wholly-owned by the Swede, officially changed its name on January 1.
>
> - o -
>
> Citing figures from the Electronic Industries Association of
>
> *Japan*, the American Electronics Association now says that the US share of worldwide electronics production fell to 39.7% in 1987 from 50.4% in 1984 while the *Japan*ese share rose to 27.1% from 23.1% over the same period and that of Europe rose to 26.4% from 23.5%, although that figure masks a decline, because the European share hit 27.6% in 1986; the newly industrialised countries of the Far East saw their 1987 share hit 6.8%, from 4.9% in 1984.
>
> [...]

In Document 3709 (ZF07-554-808) we find:

> [...]
>
> The European Community, whose Common Agriculture Policy keeps food prices high, and which has failed to persuade monopoly European airlines to reduce air fares that border on the racketeering, has now succeeded in ensuring that at times of memory chip gluts, European manufacturers that use chips in their products will not be able to buy the things at the best prices available to competitors in other parts of the world, but instead will have to bankroll manufacturers in *Japan*: the Commission has coerced 11 *Japan*ese manufacturers - Fujitsu Ltd, Hitachi Ltd, Mitsubishi Electric Corp, NEC Corp, Toshiba Corp,
>
> [Sh]arp Corp, Sanyo Denki Co, Minebea Co and Oki Electric Industry Co - to set floor prices for chips they export to Europe; the prices are between 8% and 10% above the average cost of production, weighted for each company's output; the agreements will be good for five years, and so long as the *Japan*ese makers keep prices above the floor, they will face no *dump*ing duties.
>
> [...]

This last is perhaps questionable since the topic description asks for actions taken against proven or suspected dumping, and the passage from document 3709 describes an avoidance of a dumping charge.

If the hypotheses about term distribution hold true, then documents that are listed in Figure 3.9 but do not demonstrate overlap, such as 1022, 3738, and 3697 should not be relevant. An examination of their contents reveals that one paragraph in 1022 can be considered relevant, although Japan is not mentioned specifically, nor is America or Europe:

> [...]
>
> Talking of soap operas, Toshiba appears to be doing its partner IBM and the other manufacturers involved in the dispute over the alleged *dump*ing of liquid crystal displays (CI No 1,501) no favours: the Herald Tribune quotes Takashi Shimada, top engineer in Toshiba's electron tube group, as saying "in terms of importance (to the computer system) our executives say the 1990s equivalent of the DRAM chip is liquid crystal displays" - cue more hysterical yellow perilism.
>
> [...]

but the other two have irrelevant references (e.g., dumping data onto a tape). Document 2003 presents conflicting messages: it turns out to be a series of very short newsbite, including:

> [...]
>
> Also: *JAPAN*ESE SPEECH RECOGNITION PROJECT
>
> AUDIOTEX SYSTEM BUSINESS BRISK CENTURY HIGH SCHOOL DEDICATED USERS BEMOAN QUALITY, TRAINING
>
> TECHNOLOGY *DUMP*ING IN MALAYSIA
>
> [...]

Another example topic description is shown below:

> Topic 034
>
> <dom> Domain:
>
> Science and Technology
>
> <title>Topic:
>
> Entities Involved In Building ISDN Applications and Developing Strategies to Exploit ISDN
>
> <desc> Description:
>
> Document must describe applications companies plan to build (are building, have built) for themselves or for others, which exploit ISDN's services and capabilities or identify general strategies for using ISDN.

<narr> Narrative:

To be relevant, a document must identify a company's strategy for using Integrated Services Digital Networks (ISDN) or building or using applications which take advantage of ISDN.

<con> Concept(s):

1. ISDN
2. Strategy, Applications, Products,
3. Networks

<fac> Factor(s):

<def> Definition(s):

Integrated Services Digital Networks (ISDN) - An international telecommunications standard for transmitting voice, video and data over a digital communications line.

There are 49 documents judged relevant to this topic description. By converting it to a simple TileBar query, of the form: Term Set 1: *ISDN* and Term Set 2: *application strategy*, we find TileBar descriptions like those shown in Figure 3.10. In this case it is useful to use the sorted TileBar representation. Interestingly, all of the documents in the "Both Term Sets" window, and all the documents in the "Term Set 1" window are judged to be relevant. Only document 525 in the "Term Set 2" window is relevant, and only two documents in the "Passing References" window are relevant.

These examples graphically illustrate how differences in term distribution can have different effects on relevance judgments. In topic description 034, it is important that the term *ISDN* be frequent and well-distributed throughout the text, whereas in topic description 005, both term sets needed to occur in an overlapping configuration, but in most cases in only one or two passages of the document.

These examples also show how powerful certain selected terms can be in finding the documents that have been marked as relevant. The vector space model and other similarity comparison models are designed to determine which terms are important terms automatically, usually using via inverse document frequency. In future work I plan to use the TileBar representation on vector space scores to help determine which parts of the long texts contribute to the overall vector space rankings.

3.5.2 Similarity-based Passage Retrieval Experiments

So far this chapter has focussed on the use of term distribution in passage-based retrieval. There has been a small amount of work on application of similarity-based measures to full texts; this work is discussed in this section.

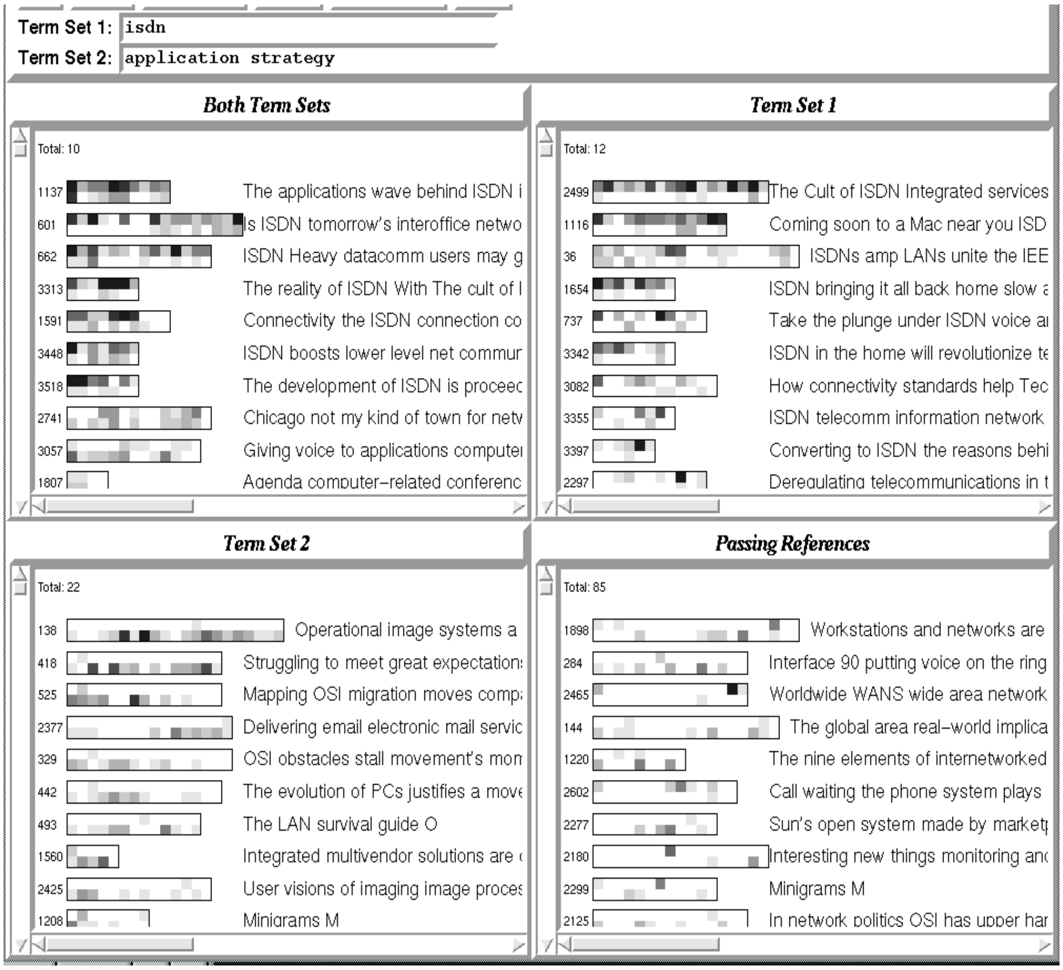


Figure 3.10: TileBars found in response to a simplified version of TREC topic description 034. Term Set 1 = *isdn* and Term Set 2 = *application strategy*.

Salton, Buckley, and Allan

One way to get an approximation to subtopic structure is to break the document into paragraphs, or for very long documents, sections. In both cases this entails using the orthographic marking supplied by the author to determine topic boundaries.

Salton, Buckley, and Allan (1991, 1993, 1994) have examined issues pertaining to the interlinking of segments of full-text documents. In the applications they have described, Salton *et al.* focus on finding subparts of a large document that either have pointers to other documents (as in "See Also" references in the encyclopedia, or replies to previously posted email messages), or are very similar in content. These links are used for the purposes automatic passage linking for hypertext. They focus more on how to find similarity among blocks of text of greatly differing length, and not so much on the role of the text block in the document that it is a part of. They find that a good way to ensure that two larger segments, such as two sections, are similar to one another is to make sure they are similar both globally and locally.

Their algorithms ensure that a document is similar to a query at several levels of granularity: over the entire text, at the paragraph level, and at the sentence level. (In this work, when applied to encyclopedia text, queries usually consist of encyclopedia articles themselves.) For two sections to be similar, they must be similar overall, at the paragraph level, and at the sentence level. To accommodate for the fact that most paragraphs differ in length, they normalize the term frequency component for the comparisons. Their results show that this procedure is more effective than using full-text information alone. This strategy, especially the sentence-level comparison, serves as a form of disambiguation, since it forces terms that have more than one sense to be used together in their shared senses. Salton *et al.* have found this approach to work quite well for the encyclopedia data, using the pre-existing See-Also links as the evaluation measure. (They point out the problems with this as an evaluation measure: since the encyclopedia is parsimonious with its reference links, many links that could reasonably be present are deliberately left out to avoid clutter.)

However, when they applied the same technique to the TREC collection, they found the results were not improved by the global/local strategy (Buckley *et al.* 1994). They attribute this to the lack of need for disambiguation among the TREC queries, since the datasets involved are more homogenous than those of the encyclopedia.

Other reasons might be that the structure of the TREC queries do not reflect the structure of the dataset, as is the case with the encyclopedia text, and that the TREC dataset is much more varied and irregular than is the encyclopedia text.

Hearst and Plaunt

An alternative approach is presented in Hearst & Plaunt (1993), which presents an experiment that demonstrates the utility of treating full-length documents as composed of a sequence of locally concentrated discussions. The strategy is to divide the documents into motivated segments, retrieve the top-scoring 200 segments that most closely match the query (according to the vector space model), and then sum the scores for all segments that are from the same document. This causes the parts of the documents that are most similar to the queries to contribute to the final score for the document. This experiment was performed on a small subset of the TREC ZIFF collection (274 documents of at least 1500 words of text each). Similarity search on segmented documents was found to perform better than full documents, and the approach of combining the scores for the top 200 segments worked significantly better than either full texts or segments alone. To be explored is the question of what portions of the documents contribute to the sum – are there several different discussions about the same subtopic, or different passages of the text corresponding to different parts of the query? Perhaps, as seen in the examples in Section 3.5.1, different explanations hold for different queries. An examination using a modified version of TileBars should help elucidate these issues.

Moffat et al.

Moffat *et al.* (1994) and Fuller *et al.* (1993) are also concerned with structured retrieval from long texts, as well as efficiency considerations required for indexing document subparts. Moffat *et al.* (1994) performed a series of experiments varying the type of document subpart that was compared and the way the subparts' were used in the ranking.

Interestingly, Moffat *et al.* (1994) found that manually supplied sectioning information may lead to poorer retrieval results than techniques that automatically divide the text. They compared two methods of dividing up long texts. The first consisted of the premarked sectioning information based on the internal markup supplied (presumably by the author) with the texts. The second used a heuristic in which small numbers of paragraphs were grouped together until they exceeded a size threshold. The results were that the small, artificial multi-paragraph groupings seemed to perform better than the author-supplied sectioning information. More experiments are necessary in this vein to firmly establish this result, but it does lend support to the conjecture that multi-paragraph subtopic-sized segments, such as those produced by TextTiling, are useful for similarity-based comparisons.

3.5.3 Other Approaches

Another recent piece of work on passage retrieval (Mittendorf & Schäuble 1994) creates a Hidden Markov Model representation of the text and the query. In order to evaluate the results the authors concatenate a sequence of abstracts (from the MEDLAR collection, which consists of 1003 abstracts and 30 queries) and try to both recognize the original boundaries of the documents as well as find the documents that are relevant to the query.

Other researchers have approximated local structure in long documents by breaking the documents into even-size pieces, without regard for any boundaries. Stanfill & Waltz (1992) report on such a technique, using the efficiency of a massively parallel computer. They divide the documents into 30-word segments and compare the queries to each segment. They also combine the scores for adjacent 30-word segments in case they break the document in an inopportune position, and then report the best n combined scores. The user can choose to see either the best sections or the heads of the best documents. This simple method, performed on texts from the Dow Jones newswire service, consisting of about 1 Gigabyte of newswires, magazines, newspapers, among others, achieves good results after extensive testing. The authors cite a precision-recall product of 0.65 on their task but do not further elaborate on this claim (it would be a challenge to accurately determine recall on such a collection unless some kind of sampling-based estimation is used).

Hahn (1990) has eloquently addressed the need for imposing structure on full-length documents in order to improve information retrieval, but proposes a knowledge-intensive, strongly domain dependent approach, which is difficult to scale to sizable text collections. Croft *et al.* (1990) describe a system that allows users direct access to structured information. Rus & Subramanian (1993) make use of certain kinds of structural information, e.g., table layout, for information extraction.

Ro (1988a) has performed experiments addressing the issue of retrieval from full texts in contrast to using controlled vocabulary, abstracts, and paragraphs alone. Performing Boolean retrieval for a set of nine queries against business management journal articles, Ro found that retrieving against full text produced the highest recall but the lowest precision of all the methods. In subsequent experiments, Ro (1988b) tried various weighting schemes in an attempt to show that retrieving against full text would perform better than against paragraphs alone, but did not achieve significant results to this effect.

3.6 Conclusions

This chapter has discussed retrieval from full-text documents. I have shown how relative term distribution can be useful information for understanding the relationship between a query and retrieved documents. I have generalized the contrast between main topics and subtopics to an analysis of all the possible combinations of term frequency and distribution between two term sets and hypothesized about the usefulness of each distributional relationship.

I have also introduced a new display device, called TileBars, that demonstrates the usefulness of explicit term distribution information. The representation simultaneously and compactly indicates relative document length, query term frequency, and query term distribution. The patterns in a column of TileBars can be quickly scanned and deciphered, aiding users in making fast judgments about the potential relevance of the retrieved documents. TileBars can be sorted according to their distribution patterns and term frequencies, aiding the users' evaluation task still more. Two queries from the TREC collection were analyzed using TileBars and it was shown that the relevant documents for each query demonstrated radically different patterns of distribution of the chosen query terms.

Currently only two term sets are contrasted at a time; this can be easily extended to three or four. It is most likely the case that any more than four term sets will make the representation difficult to interpret. Another extension to be made to the existing implementation of TileBars is improvement of the simple pattern sorting heuristic. Studies should to be done to determine what kinds of pattern sortings are most informative. In the future the Tilebars should also be evaluated in terms of their use in relevance feedback and with respect to how users interpret the meaning of the term distributions. The analysis should compare users' expectations about the meaning of the term distributions against the analysis shown in the distribution chart. It may be useful to determine in what situations the users' expectations are not met, in hopes of identifying what additional information will help prevent misconceptions.

Information access mechanisms should not be thought of as retrieval in isolation. Cutting *et al.* (1990) advocate a text access paradigm that "weaves together interface, presentation and search in a mutually reinforcing fashion"; this viewpoint is adopted here as well. For example, the user might send the contents of the Passing References window of a TileBar session to a Scatter/Gather session (Cutting *et al.* 1993), which would then cluster the documents, thus indicating what main topics the passing references occurred in. The user could select a subset of the clusters to be sent back to the TileBar session. This kind of integration will be attempted in future work.

Chapter 4

Main Topic Categories

4.1 Introduction

This chapter presents an algorithm that automatically assigns multiple main topic categories to texts, based on computing the posterior probability of the topic given its surrounding words, without requiring pre-labeled training data or heuristic rules. The algorithm significantly outperforms a baseline measure and approaches the levels of inter-indexer consistency displayed by nonprofessional human indexers. The chapter also describes the construction of a general category set from an existing hand-built lexical hierarchy.

The approach to categorization described here is one in which only the simplest assumptions are made about what it means to categorize the contents of a text. This is done for the purposes of robustness, scalability, and genre transferability. More reasonable results could be obtained from more structured and domain-specific analyses of the text, but at the cost of not allowing for wide applicability.

4.2 Preview: How to use Multiple Main Topic Categories

The capability to automatically assign main topic labels (in this and the next chapter, the terms "categories", "main topics", and "labels" are used interchangeably) leads to a new paradigm for browsing the contents of full-length texts: the labels can be used to help *contextualize* the results of a query; i.e., show the user the topics that characterize the documents associated with the results of a query. In Chapter 5, I explore the hypothesis that users need more contextual information when dealing with full-length texts than with abstracts and short text, in part because similarity information is less useful when comparing lengthy documents. Here I present one example of this idea.

If the results of a user's query are situated with respect to the main topics of the documents, a user with only a vague notion of what context the term should appear in can browse the output of the categorizer to find appropriate texts. For example, Figure 4.1 shows automatically assigned main topic categories for five texts from the ACL/DCI collection

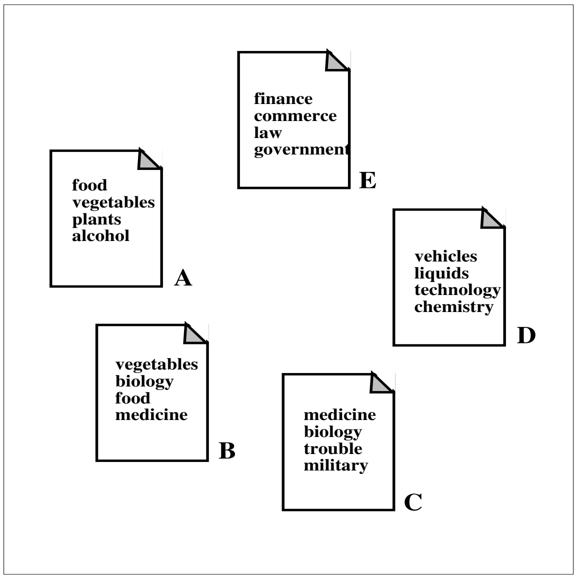


Figure 4.1: Main topic categories assigned by the algorithm described in this chapter to texts that contain the term "contaminant" at least twice in a small corpus of newspaper text.

(Church & Liberman 1991) of articles from the *Wall Street Journal* in which the string "contaminant" occurs at least twice. Glancing at these we can get a feeling for the "gist" of each article. For example, documents A and B are assigned categories relating to food, while document C is assigned two very different categories – medicine and military – because the article discusses the accidental release of an agent for biological warfare and the subsequent medical damage control efforts. Document D discusses contaminants in a technical context while document E discusses contaminants in a financial context; in other words, rather than focusing on the medical or environmental aspects of a contamination, it focuses on associated business and legal costs.

Note that this example, especially document C, highlights another point: texts, especially long texts, are not always best represented as one topic from one semantic class. Rather they are often about two or more themes and some relationship among these. Thus classifying documents strictly within a topic hierarchically can be misleading, because the multiple themes that co-exist are not necessarily ones that are commonly considered to be in the same semantic frame. These and related issues are discussed in greater detail in Chapter 5.

This chapter is structured as follows: Section 4.3 describes the categorization algorithm, Section 4.4 presents an evaluation of the algorithm, and Section 4.5 describes the way the general thesaurus-like category set was acquired.

4.3 Automatic Assignment of Multiple Main Topics

This chapter describes a mechanism for assigning multiple main topics to lengthy expository texts. The algorithm is a modification of a disambiguation algorithm described in Yarowsky (1992). It requires a training phase that determines which terms should be weighted highly as evidence for each category. The training does not require pre-labeled texts; rather it relies on the tendency for instances of different categories to occur in different lexical contexts to separate the evidence. When assigning topics to a text, the algorithm measures how much evidence is present for every category; the categories with the most evidence are considered to be the main topic categories of the text.

The categorization algorithm is statistical in nature and is based on the assumption that main topics of a text are discussed throughout the length of the text. Thus although it looks at the evidence supplied by individual lexical items, it does not take the structure of the text into account, e.g., how the lexical items are related to one another syntactically or by discourse structure. The algorithm is successful at identifying schema-like categories; it identifies terms associated with the categories that are not necessarily originally specified as members of the categories. However, because the categories are pre-defined, the algorithm cannot recognize or produce novel labels. For this reason, the results of the categorization algorithm should be used in conjunction with terms that occur frequently throughout the text when characterizing the texts' content. Fixed categories should play only a partial role in the characterization of the contents of the text.

This chapter also discusses an approach to creating thesaurus-like categories from an existing hand-built lexicon, WordNet (Miller *et al.* 1990). The first step is an algorithm for breaking up the WordNet noun hierarchy into small groups of related terms, and the second step determines which groups to combine together in an attempt to create schema-like categories. This step uses lexical association information from a large corpus to determine which groups are most similar to one another. This procedure yields a set of categories that can then be used as general category labels for lengthy expository texts.

4.3.1 Overview

The category assignment algorithm works as follows. A measure of association between words and categories is found by training on a large text collection; the training algorithm is described in the following sections. This measure of association is used to characterize the words of the document to which categories are to be assigned. The algorithm looks up how strongly associated each word in the text is with all of the categories in the category set. The scores for each category are added together, and the top scoring categories, subject to a user-specified cutoff, are reported after the entire document has been processed. The association measure is a normalization of $P(C_i|w_j)$ as shown below.

Earlier I experimented with algorithms that tried to determine which sense (category) of a word was being used before allowing that word to contribute to evidence for the overall categorization of the algorithm. This requires using a window of words surrounding a word to determine which sense of the target term is in use (using a variation of Yarowsky's (1992) algorithm, see below). Although I abandoned this approach, the actual current implementation processes whole windows of words at a time, in effect periodically "probing" the document. Depending on the window size and the frequency of the probes, this can result in counting each term a constant number of times, rather than one time only, but this does not change the resulting ranking of the categories.

4.3.2 Yarowsky's Disambiguation Algorithm

The topic assignment algorithm described here is a modification of a disambiguation algorithm described in Yarowsky (1992). Yarowsky defines word senses as the categories listed for a word in *Roget's Thesaurus* (Fourth Edition), where a category is something like TOOLS/MACHINERY. For each category, the algorithm

> 1. Collects contexts that are representative of the Roget category
> 2. Identifies salient words in the collective contexts and determines weights for each word, and
> 3. Uses the resulting weights to predict the appropriate category for a polysemous word occurring in a novel text. (Yarowsky 1992)

In other words, the disambiguation algorithm assumes each major sense of a homograph is represented by a different thesaurus-like category. Therefore, an algorithm that can determine which category an instance of a term belongs to can in effect disambiguate the term. The disambiguation is accomplished by comparing the terms that fall into a wide window surrounding the target term to contexts that have been seen, in a training phase, to characterize each of the categories in which the target term is a potential member. The training phase determines which terms should be weighted highly for each category, using a mutual-information-like statistic. The training does not require pre-labeled texts; rather it relies on the tendency for instances of different categories to occur in different lexical contexts to separate the senses. After the training is completed a word is assigned a sense by combining the weights of all the terms surrounding the target word and seeing which of the possible senses that word can take on has the highest weight.

I extend this algorithm to the text categorization problem as follows. Instead of choosing from the set of categories that can be assigned to a particular target word, this new version of the algorithm measures how much evidence is present for *all* categories, independently of what word occurs in the center of the context being measured. After the entire document has been processed, the categories with the most evidence are identified as the main topic categories of the text. This algorithm is based on the assumption, discussed in Chapter 3, that main topics of a text are discussed throughout the length of the text. The algorithm is described in more detail in the next two subsections.

4.3.3 Lexically-Based Categories

For the purposes of this algorithm, a category is defined by the set of lexical items that comprise it. The implementation uses a category set derived from WordNet (Miller *et al.* 1990), a large, hand-built online repository of English lexical items organized according to several lexico-semantic relations. The implementation does not use *Roget's* categories because at the time of writing they are no available to the public in electronic form. The algorithm used to derive the WordNet-based categories is described in Section 4.5, with the goal of achieving wide coverage using general categories. A moderate size category set was used in order to facilitate comparisons against judgements made by human subjects (who would be overwhelmed by too large a category set).

The algorithm works by automatically determining, for each category, which lexical items tend to indicate the presence of that category. The evidence for presence of a category is determined by not only by the presence of the lexical items that make up the category, but also by terms that have been found to co-occur in a salient manner with the category terms (described in detail below). For example, the "vehicles" category, consisting of names of kinds of vehicles, could be indicated by terms indicating where vehicles are used, e.g., "road", "ocean", etc. Ideally, the category itself might contain terms that indicate the semantic frame in which the category is used. For example, the "athletics" category contains terms about athletes, playing fields, and sports implements, as well as names of sports. It is difficult to determine where to draw a line between category-specific terms and terms that occur more generally. However, the algorithm helps decide this by indicating which terms outside the category nevertheless co-occur with it significantly and to the exclusion of other categories. Thus, in some cases the algorithm discovers terms that support the frame-like meaning of the categories.

Sparck-Jones (1971) discusses at length the difference between synonyms and semantically related terms in a category definition. For example, terms grouped with "desire" in *Roget's Thesaurus* include "wish", "fancy", and "want", which can be called synonyms. In contrast, terms grouped with "navigation" include "boating", "oar", and "voyage"; these are not synonyms but are semantically, associationally related to the navigation schema. She concludes in Sparck-Jones (1986) that the semantic-based classes are more effective for information retrieval, although does not claim to verify this rigorously.

4.3.4 Determining Salient Terms

Yarowsky 1992 defines a salient word as "one which appears significantly more often in the context of a category than at other points in the corpus" (p 455). For example, the term "lift" can be salient for the machine sense of "crane" but not for the bird sense. He formalizes this with a mutual-information-like estimate: $P(w|RCat)/P(w)$, the probability of a word w occurring in the context of the Roget category $RCat$ divided by the probability of the term occurring in the corpus as a whole. Yarowsky notes that $P(w|RCat)$ can be computed by determining the number of times w occurs in the context surrounding terms that are members of $RCat$. He notes, however, that this estimate is unreliable when w is infrequent in the corpus, and corrects for this by employing a smoothing algorithm described in Gale *et al.* (1992b).

Once all of the computations for $P(w|RCat)$ have been computed, disambiguation can take place. Yarowsky combines the evidence supplied by the words surrounding an instance of the word that is being disambiguated as follows:

$$\underset{RCat}{\text{Argmax}} \sum_{w \ in \ context} \log \frac{P(w|RCat)P(RCat)}{P(w)}$$

where a context of 50 words is allotted on either side of the target word (addition is used since the formula takes the logs of the evidence weights). Yarowsky assumes a uniform distribution for $P(RCat)$, and notes that $P(w)$ can be omitted as well since it will not effect the results of the maximization.

This algorithm does not enforce mutual exclusivity on evidence for different categories, although weight assigned to one category does detract from weight that can be assigned to any other category (since the frequency of co-occurrence of a word with a category member is divided by the overall frequency of the word). The lack of mutual-exclusivity is useful in that it allows one word to provide partial evidence for multiple categories.

Training proceeds by first collecting global frequency counts over a corpus. For the current implementation, training was done on *Grolier's American Academic Encyclopedia* ($\approx$ 8.7M words). In the current implementation of the algorithm, terms are checked against WordNet (Miller *et al.* 1990) in order to place them in a "canonicalized" form. Words that are listed on a 454-word "stoplist," (i.e., a list of closed-class words and other highly frequent words) are not used in the calculation of evidence for category membership.

If a pair of adjacent words matches a compound contained in WordNet, then that pair is considered a term, instead of as the individual words that comprise it. If the word does not participate in a two-member compound, then its membership in WordNet alone is investigated. If this check fails, then the term's inflections are removed using a modified version of WordNet's morphological analyzer, and the stemmed version is looked up in WordNet. In case of failure, the next two modifications are conversion of the term's characters to lowercase and reapplication of morphological stemming. If all else fails, the word is recorded in its original form.

As mentioned above, Yarowsky's algorithm is designed to perform word disambiguation. After training has been completed, the term weights can be used to classify a new instance of a term that is a member of one or more categories into the category with the most contextual evidence. However, category assignment is computed somewhat differently.

Evidence for category membership is determined by evaluating co-occurrence information within a fixed-length window of terms surrounding each instance of a target term. To prevent the evidence supplied by frequent terms from dominating the evidence supplied by infrequent terms, the evidence contributed by a particular member of a category is normalized by the number of times that term occurs in the corpus as a whole. (Yarowsky (1992) also weights the terms in this manner), thus requiring two passes through the training data. Instead of smoothing estimates for infrequent terms, this implementation simply excludes infrequent terms (terms whose frequency is less than a threshold) from contributing evidence. This is done because it is unlikely that a very infrequent term will be able to provide reliable evidence for the category, and thus it is excluded from making any kind of contribution at all.

In the current implementation of the algorithm a term window surrounding an instance of a word that is a member of a category is counted equally as evidence for all of the categories of which the word is a member. However, it would be interesting to incorporate this and determine whether or not re-estimation caused results to improve. Another possibility is to use the re-estimation step to adjust the bias of the algorithm to a corpus different than the one initially trained on.

Another issue is that of context window size. Gale *et al.* (1992b) find that sometimes words even 10,000 positions away from the target term are useful for training; Yarowsky (1992) uses a fixed window of 100 words. That window size was also used in the current implementation of the algorithm; however, a more meaningful way to specify the context window would be to use coherent multi-paragraph units as discovered by the TextTiling algorithm of Chapter 2. The re-estimation algorithm can also play a role in determining an appropriate window size, as follows (see Figure 4.2). The tiling algorithm, using only term repetition, determines the windows to be used as input to the training phase of the categorization algorithm. The categorization algorithm is then used to assign disambiguated labels to many of the terms which are then re-input to the tiling algorithm, which presumably can now generate more accurate tile information, and so on. The training loop idea has not yet been implemented.

In the training phase of the current implementation, if a word is a member of a category, then that word is not allowed to count as evidence for the category. This makes more sense for the disambiguation algorithm than for the topic labeling algorithm, but in both cases the word should be able to count as evidence for the category it is a member of, according to some prior probability of its tendency to represent that category versus any other category of which it may be a member.

4.3.5 Related Work and Advantages of the Algorithm

There exist other systems in which multiple categories are assigned to documents, e.g., Masand *et al.* (1992), Jacobs & Rau (1990), Hayes (1992). However, unlike the method suggested here, these systems require large volumes of pre-labeled texts in order to perform these classifications. Larson (1992), Larson (1991) presents an algorithm that automatically assigns Library of Congress Classification numbers to bibliographic records consisting of titles and subject headings after forming clusters based on training from existing records. The method works well but requires pre-defined subject headings as attributes for classification.

The approach of Liddy & Paik (1992) is most similar to that presented here. Liddy &

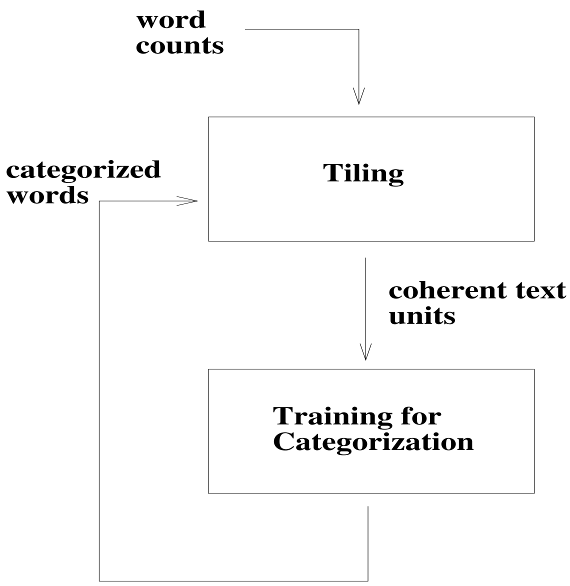


Figure 4.2: A proposed training loop: TextTiling, using only term repetition information, provides context window information for the categorization algorithm which then supplies term category information to improve the results of TextTiling, which aids in the reestimation of the priors for the categorization algorithm, and so on.

Paik (1992) use Subject Code assignments from the LDOCE dictionary, creating in effect a set of general categories. The algorithm presented here makes a probabilistic estimate of the likelihood of a category given the terms that occur. In contrast, the system of Liddy & Paik (1992) uses heuristics to determine word senses based on how many words that can be assigned a particular code occur in a sentence, as well as how likely it is for the candidate codes in the sentence to co-occur. Thus it also does not require pre-labeled texts but it does require a large number of words to have been assigned to categories in advance. The categorization algorithm described here also requires some terms to be assigned to each category in advance, but it automatically chooses additional terms from the corpus to act as strong indicators for each category. Thus it should be more adaptable to new category sets, that is, category sets that characterize specialized domains such as academic computer science. It would be useful to run an experiment comparing the results of the two algorithms.

Other categorization algorithms also deal with the issue of choosing salient features. Lewis (1992) defines *feature selection* as the process of choosing, for each category, a subset of terms from an indexing language to be used in predicting occurrences of that category. He uses a mutual information statistic within a probabilistic framework, choosing the highest scoring terms for each category to act as indicators for the presence of that category. This approach to term weighting is the most similar to that described here.

Many knowledge-based classification systems also recognize the need to determine which terms outside of those already specified are good indicators of the category. Riloff & Lehnert (1992) use a pre-labeled training set to extract defining features from a frame-like representation for each document, in a framework in which a single phrase can be enough to indicate the presence of a category. Jacobs (1993), in a framework combining knowledge-based and statistical information, explores several different ways to determine good indicators, including weighting terms according to a mutual information statistic, using exception lists, and finding terms that tend to surround category terms but are not a part of the category themselves. Fung *et al.* (1990) using a probabilistic network for categorization, requires users to select which features from a set indicate the relevance of training documents and then automatically determines the weights to place on the links. The disadvantage to all these approaches is that they require pre-labeled texts or user judgments for the training step.

Many algorithms have been developed that use co-occurrence information for determining category membership or to build thesaurus classes. Crouch (1990),Grefenstette (1992),Salton (1972),Sparck-Jones (1986), and Ruge (1991) all use co-occurrence information derived from corpora to determine how to expand queries with related terms, and show that this information can improve retrieval. (But see Peat & Willett (1991) for a criticism of this kind of approach.) Deerwester *et al.* (1990) use co-occurrence terms among documents (compressed with multivariant decomposition) to determine semantic relatedness among documents. Co-occurrence information has been found to be useful for a variety of tasks in computational linguistics as well (e.g., Church & Hanks (1989), Smadja & McKeown (1990), Justeson & Katz (1991)).

An advantage of the Yarowsky weighting scheme is that it uses co-occurrence information to classify terms into pre-defined, intuitively understandable classes, as opposed to classes derived from the data. Although categories or classes derived from data are useful for many kinds of applications, intuitive categories may be more appropriate when interfacing between the system and the user. This supposition is visited in more detail in Chapter 5.

Another advantage of the algorithm is that it can accommodate multiple category sets. Categorization algorithms based on clustering can only present one view on the data, based on the results of the clustering algorithm, but as shown above, documents may be similar on only one out of several main topic dimensions. Algorithms that train on pre-labeled texts can also represent multiple simultaneous categories, but are confined to using only the category sets that have been pre-assigned (since in most cases thousands of pre-labeled documents are necessary to train these algorithms).

4.4 Evaluation of the Categorization Algorithm

A common way to evaluate a categorization algorithm is to compare its labelings with those assigned by human categorizers. For some test collections a "correct" set of labels already exists, and the program's results can be measured directly against these. For example, Masand *et al.* (1992), Jacobs (1993), and Hayes (1992) use human-assigned labels both for training and judging their classification systems.

When there exists a large training base of examples, it is safer to assume that comparing against one judgment per document is accurate. However, because no large training set is available for this task, and because inter-indexer consistency tends to be low (Bates 1986), a better evaluation metric is to compare the inter-indexer consistency of the algorithm with that of human judges. (Inter-indexer consistency is the average number of category assignments a judge makes in common with the other judges for a particular document.) Furthermore, Cooper (1969) shows that in a restricted case at least, increased inter-indexer consistency leads to increased expected search effectiveness, and Rolling (1981) provides more supporting evidence to this effect.

Following Gale *et al.* (1992a), the performance of the algorithm is evaluated against both a lower bound and an upper bound. The lower bound represents the minimal performance that any algorithm ought to be able to surpass. Often this boundary is what would result if an algorithm always made the most likely choice, e.g., for a part-of-speech tagger, a lower bound might be the percentage correct obtained by always assigning the most likely part-of-speech category for each word. Useful lower bounds are not always easily formulated; sometimes an algorithm's results should just be compared against what an algorithm making random assignments would produce (surprisingly, it is not infrequent that proposed algorithms do not perform much better than chance). Since no priors on category assignments are available for evaluation of the category set described here, the lower bound or baseline in this evaluation is the performance of an algorithm making random choices. Often in computational linguistics algorithms the upper bound is that of human performance; the algorithm should not be expected to do better than a human would on a task with a goal of matching human intuitions. In this evaluation, human inter-indexer indexing, as described above, is the upper bound for evaluation.

4.4.1 The Test Set

The texts used in the evaluation experiments were chosen to satisfy several desiderata. They are:

- lengthy, but short enough for human judges to skim several of them in a reasonable amount of time,
- general in subject matter in order to match the the test category set,
- varied in terms of main topic subject matter, so that a variety of categories will be assigned, and
- publically accessible to facilitate comparison against other categorization methods.

The Brown Corpus satisfies these criteria; this experiment uses the first 300 articles.[1] Each document is approximately 190 lines long (or 2030 words, on average) and in most cases consists of an unbroken stream of text.[2] The texts of the documents are cut off after the first 190 lines, so in most cases readers do not see the entire text.

4.4.2 The Experiment

Out of these 300 articles, 10 were chosen at random. The 10 texts were separated into two groups (labeled A and B) of 5 texts each, in order to reduce the reading load on the judges. Each judge was given the list of 106 categories and the five texts from either group A or group B, and the following instructions:

> *I'd like you to look over the categories briefly, and then read quickly or skim each text. Each time after you read a text, look at the category list again and choose the five best categories to describe the text's main topic(s). List the categories in ranked order, with best first. Use the category number, and please include at least the beginning of the category name so I know you didn't put the wrong number by accident. The text name occurs at the beginning of each file.*

The judges did not know that their rankings would be compared against those generated by a computer program.

Ten sets of judgments were collected; five for each group of texts. There is disagreement in the literature about how to compute inter-indexer consistency (e.g., Rolling (1981),Henzler (1978)), however, in most cases this is done in a pairwise manner. We are interested in how closely the program matches the human judgments on average.

[1]Document numbers A.01-F.48, H.01-H.30, J.01-J.80.

[2]A few documents consisted of several distinct articles combined, the first blending directly in to the next.

4.4.3 Analysis of Results

The inter-indexer agreement was computed for each judge and each document; that is, the average number of category assignments the judge made in common with the other judges for a particular document. More formally, if there are m judges making k choices for each document,

$$score(j_d) = \left(\sum_{1 \leq i \neq j \leq m} c(j_d, i_d) \right) / (m-1) * k$$

where j_d is the list of five categories assigned by judge j to document d, and $c(j_d, i_d)$ is the number of categories assigned to document d by both judge j and judge i. (This is equivalent to taking the average of the pairwise scores.) Note that for this calculation, relative ranking of categories is not taken into account.

Table 4.1 shows the categories chosen by the judges and the algorithm for two of the test documents. The labeling of document A.08 had high inter-indexer consistency both among judges and the algorithm. For document E.25, the algorithm did not rank *medicine*, the judges' highest category, in its top five (rather, it was ranked eighth), although there is strong agreement among the other terms; this was an exceptional case (see below). The document in question discusses research advances on technology to be used in a medical context.

judge A	judge B	judge C	judge D	judge E	Algorithm
33 government	34 politics	102 actions	34 politics	33 government	33 government
32 legal system	33 government	104 happening	33 government	34 politics	36 finance
34 politics	37 work	34 politics	104 happening	36 finance	32 legal_system
36 finance	102 actions	37 work	06 cities	32 legal_system	35 commerce
37 work	36 finance	59 information	29 conflict	29 conflict	29 conflict

judge F	judge G	judge H	judge I	judge J	Algorithm
27 medicine	27 medicine	27 medicine	27 medicine	25 body_process	87 light
02 measure	44 technology	44 technology	02 measure	27 medicine	44 technology
45 electronics	45 electronics	02 measure	44 technology	44 technology	45 electronics
44 technology	52 science	87 light	45 electronics	45 electronic	53 physics
99 defense	71 cell_biology	26 body_parts	26 body_parts	100 stuff	66 machines

Table 4.1: Category assignments to two documents (A.08 and E.25) by human judges and by the algorithm.

One way to evaluate the results of an algorithm is to compare its performance against a baseline. In this case, we compute the expected inter-indexer consistency score of an algorithm that chooses from the category set at random. This baseline is computed as follows, if we are not concerned with relative ranking of categories. The model is one of chosing k categories without replacement from a set of n unique categories, where each choice of category is independent from the previous and subsequent choices. The underlying distribution is assumed to be hypergeometric (sampling without replacement).

The algorithm is required to choose k categories, and its choices are compared against those of one judge, who is assumed to be correct in all k choices. The number of ways to choose i categories correctly out of k choices, from a set of n categories without replacement (if order does not matter), is $\binom{k}{i}\binom{n-k}{k-i}$.

When i categories are identified correctly, the percentage correct is i/k; therefore the expected percentage correct when comparing against one judge for a random category assignment is $\sum_{i=1}^{k} \frac{i}{k}\binom{k}{i}\binom{n-k}{k-i} \Big/ \binom{n}{k}$.

In this experiment, $k = 5$ and $n = 106$. Substituting in these values we determine that the expected percent correct for a random choosing process is 5%. The variance is $E(score^2) - (E(score))^2 = 0.01 - .0025 = .0075$.

Since we have five independent comparisons, the average score (the mean of the average) is the sum of the five means divided by five, or the mean of any one (0.05). The variance of the average is the sum of the variances divided by k^2, 0.0015. Thus we have a Gaussian random variable with mean 0.05 and variance 0.0015. This means a score greater than 13% (two standard deviations greater than the mean) happens less than 5% of the time if the categories are chosen at random.

Table 4.2 presents summary data for the judges and two ways of scoring the output of the algorithm. The judges' average consistency score is 54%; the algorithm when restricted to its top 5 choices has a consistency score of 39% and when allowed to present its top 7 choices has an average score of 52%. Thus our results perform much better than the baseline, since on average the algorithm matches 39% (1.96/5) of the judges' choices.

	Average for Judges	Average for Algorithm-5	Average for Algorithm-7
Group A	0.54	0.39	0.50
Group B	0.54	0.39	0.53
Average	0.54	0.39	0.52

Table 4.2: Overall inter-indexer consistency scores. *Algorithm-5* indicates the score for the algorithm's five top-ranked categories, while *Algorithm-7* indicates the score for the algorithm's top seven categories.

Tables 4.3 and 4.4 present these results in more detail. Table 4.3 shows the percentage of inter-indexer agreement for each document and for each judge in group A, as well as the average consistency over all judges for each document. Table 4.4 shows the corresponding information for group B.

These tables indicate the percentage agreement between the program's scores and those of the judges. Note that when comparing a judge against the other judges, comparisons are made against four other rankings, but when comparing the program against the judges, comparisons are made against all five judges. (Including the program's scores when comparing judges against judges would bias the comparison to favor the program by giving its assignments equal weight to the judges' assignments. Excluding the program's assignments is a more stringent test of its accuracy.)

These tables also list two rows of scores for the program. As above, *Algorithm-5* shows the inter-indexer consistency when only the top five categories chosen by the program are used in the comparison, thus making a fair comparison against the scores of the judges. *Algorithm-7* shows the percentage agreement when the top seven categories generated by the program are used in the comparison to the judge's top five categories, thus improving recall at the expense of precision. Although this number cannot be directly compared against the scores of the human judges, it does show that if the algorithm is allowed to include a few extra categories, it will indeed bring in more relevant categories.

judge	A.08	C.10	F.32	J.11	J.21
A	0.60	0.55	0.55	0.65	0.60
B	0.40	0.30	0.60	0.60	0.45
C	0.40	0.55	0.55	0.60	0.35
D	0.45	0.50	0.60	0.60	0.60
E	0.55	0.40	0.50	0.55	0.40
Average	0.48	0.46	0.66	0.60	0.48
Algorithm-5	0.44	0.48	0.36	0.20	0.48
Algorithm-7	0.64	0.56	0.48	0.20	0.60

Table 4.3: Inter-indexer consistency scores for each judge on each document in group A.

judge	B.04	E.25	J.10	J.15	J.35
F	0.50	0.65	0.45	0.65	0.60
G	0.45	0.45	0.45	0.60	0.45
H	0.60	0.55	0.55	0.60	0.65
I	0.50	0.70	0.50	0.60	0.55
J	0.35	0.55	0.45	0.55	0.35
Average	0.48	0.60	0.48	0.60	0.52
Algorithm-5	0.44	0.40	0.28	0.60	0.24
Algorithm-7	0.44	0.48	0.48	0.72	0.52

Table 4.4: Inter-indexer consistency scores for each judge on each document in group B.

Looking more carefully at the tables, we see the algorithm performed most poorly on documents J.10, J.11, and J.35 when restricted to the top five categories. A similar problem occurred with both J.10 and J.11. The top-ranked category for J.10 for both the program and the indexers is *bugs/insects*. Similarly, the top-ranked category for J.11 for both the algorithm and four of the indexers is *reptiles/amphibians*. In both cases, the judges marked as less important other categories such as *measure* and *science*. By contrast, in both cases the algorithm lists only other animal categories. Upon reflection this behavior is not surprising since animal terms will tend to occur in similar contexts in training, and since the categories were not trained to be mutually exclusive.

Another way to measure the results is to determine how often the program assigns the most important categories. Overall, the program's performance was strong with respect to choosing highest-ranked categories.

If a majority ($\geq$ 3) of the judges agree on the top-ranked category, this category is called the majority top choice. A category that is ranked highest by at least one judge is referred to as a minority top choice. Eight out of the ten documents had majority top choices. Of these, in four cases, the program's top choice was the majority top choice. In two cases, the program's top choice matched a minority top choice. In one of the remaining cases, the program was off-by-one, ranking the majority top choice second, and in the other, the majority top choice was the program's eighth choice. In the two remaining cases in which no majority existed, the program's top choice matched a minority top choice.

4.5 Creating Thesaural Categories

Recently, much effort has been applied to the creation of lexicons and the acquisition of semantic and syntactic attributes of the lexical items that comprise them, e.g, Alshawi (1987), Calzolari & Bindi (1990),Grefenstette (1992),Hearst (1992), Markowitz *et al.* (1986), Pustejovsky (1987), Schütze (1993a), Wilks *et al.* (1990). However, a lexicon as given may not suit the requirements of a particular computational task. Lexicons are expensive to build; therefore, it is preferable to adjust an existing one to meet an application's needs over creating a new one from scratch. This section describes a way to add associational information to a hierarchically structured lexicon in order to create thesaurus-like categories useful for the topic assignment task.[3]

One way to label texts, when working within a limited domain of discourse, is to start with a pre-defined set of topics and specify the word contexts that indicate the topics of interest, as in Jacobs & Rau (1990). Another way, assuming that a large collection of prelabeled texts exists, is to use statistics to automatically infer which lexical items indicate which labels, as in Masand *et al.* (1992). In contrast, the goal here is to assign labels to general, domain-independent text, without benefit of pre-classified texts. In all three cases, a lexicon that specifies which lexical items correspond to which topics is required. The topic labeling method of this chapter is statistical and thus requires a large number of representative lexical items for each category.

Because a good, large, online public-domain thesaurus is not currently available, this section describes a way to derive one from a hierarchical lexicon. The starting point for the thesaurus is WordNet (Miller *et al.* 1990), which is readily available online and provides a large repository of English lexical items. WordNet[4] is composed of *synsets*, structures containing sets of terms with synonymous meanings, thus allowing a distinction to be made

[3]Much of the work in this section appeared in a similar form in Hearst & Schütze (1993).

[4]All work described here pertains to Version 1.3 of WordNet.

between different senses of homographs. Associated with each synset is a list of relations that the synset participates in. One of these, in the noun dataset, is the hyponymy relation (and its inverse, hypernymy), roughly glossed as the "ISA" relation. This relation imposes a hierarchical structure on the synsets, indicating how to generalize from a subordinate term to a superordinate one, and vice versa.[5] This is a very useful kind of information for many tasks, such as reasoning with generalizations and assigning probabilities to grammatical relations (Resnik 1992).

This lexicon must be adjusted in two ways in order to facilitate the label assignment task. The first is to collapse the fine-grained hierarchical structure into a set of coarse but semantically-related categories. These categories will provide the lexical evidence for the topic labels. (After the label is assigned, the hierarchical structure can be reintroduced.) Once the hierarchy has been converted into categories, the categories can be augmented with new lexical items culled from free text corpora, in order to further improve the labeling task.

The second way the lexicon must be adjusted is to combine categories from distant parts of the hierarchy. Of particular interest are groupings of terms that contribute to a frame or schema-like representation (Minsky 1975); this can be achieved by finding associational lexical relations among the existing taxonymic relations. For example, WordNet has the following synsets: "athletic game" (hyponyms: baseball, tennis), "sports implement" (hyponyms: bat, racquet), and "tract, piece of land" (hyponyms: baseball_diamond, court), none of which are closely related in the hierarchy. We would like to automatically find relations among categories headed by synsets like these. (In Version 1.3, the WordNet encoders have placed some associational links among these categories, but still only some of the desired connections appear.)

In other words, links should be derived among schematically related parts of the hierarchy, where these links reflect the text genre on which text processing is to be done. Schütze (1993b) describes a method, called WordSpace, that represents lexical items according to how semantically close they are to one another, based on evidence from a large text corpus. To create structured associational information, the term-similarity information from WordSpace is combined with the category information derived from WordNet to create schema-like super-categories.

The next subsection describes the algorithm for converting the WordNet hierarchy into a set of categories. This is followed, in subsection 4.5.2 by a discussion of how these categories are to be used and why they need to be improved. Subsection 4.5.3 describes how WordSpace an be used to bring disparate categories together to form schematic groupings while retaining the given hierarchical structure.

[5]Actually, the hyponomy relation is a directed acyclic graph, in that a minority of the nodes are children of more than one parent. I will at times refer to it as a hierarchy nonetheless.

4.5.1 Creating Categories from WordNet

An algorithm is needed to decompose the WordNet noun hierarchy into a set of disjoint categories, each consisting of a relatively large number of synsets, creating categories of a particular average size with as small a variance as possible, where each category consists of a relatively large number of synsets (this is necessary for the text-labeling task, because each topic must be represented by many different terms). There is some limit as to how small this variance can be because there are several synsets that have a very large number of children (there are sixteen nodes (synsets) with a branching factor greater than 100). This primarily occurs with synsets of a taxonymic flavor, i.e., mushroom species and languages of the world. There are two other reasons why it is not straightforward to find uniformly sized, meaningful categories:

(i) There is no explicit measure of semantic distance among the children of a synset.

(ii) The hierarchy is not balanced, i.e., the depth from root to leaf varies dramatically throughout the hierarchy, as does the branching factor. (The hierarchy has ten root nodes; on average their maximum depth is 10.5 and their minimum depth is 2.)

Reason (ii) rules out a strategy of traveling down a uniform depth from the root or up a uniform height from the leaves in order to achieve uniform category sizes.

For the purposes of the description of this algorithm, a synset is a node in the hierarchy. A descendant of synset N is any synset reachable via a hyponym link from N or any of N's descendants (recursively). This means that intermediate, or non-leaf synsets, are also classified as descendants. The term "child" refers to an immediate descendant, i.e., a synset directly linked to N via a hyponym link, and "descendant" to indicate linkage through transitive closure.

The algorithm used here is controlled by two parameters: upper and lower bounds on the category size (see Figure 4.3). For example, the result of setting the lower bound to 25 and the upper bound to 60 yields categories with an average size of 58 members. An arbitrary node N in the hierarchy is chosen, and if it has not yet been marked as a member of a category, the algorithm checks to see how many unmarked descendants it has. In every case, if the number of descendants is too small, the assignment to a category is deferred until a node higher in the hierarchy is examined (unless the node has no parents). This helps avoid extremely small categories, which are especially undesirable.

If the number of descendants of N falls within the boundaries, the node and its unmarked descendants are bundled into a new category, marked, and assigned a label which is derived from the synset at N. Thus, if N and its unmarked descendants create a category with k members, the number of unmarked descendants of the parent of N decreases by k.

If N has too many descendants, that is, the count of its unmarked descendants exceeds the upper bound, then each of its immediate children is checked in turn: if the child's descendant count falls between the boundaries, then the child and its descendants are bundled into a category. If the child and its unmarked descendants exceed the upper bound,

```
for each synset N in the noun hierarchy
  a_cat(N)

  a_cat(N):
  if N has not been entered in a category
      T <- #descendents(N)

      if ((T >= LOWER_BRACKET)
            && (T <= UPPER_BRACKET))
         mark(N,NewCatNumber)

      else if (T > UPPER_BRACKET)

        for each (direct) child C of N
           CT <- #descendents(C)
           if ((CT >= LOWER_BRACKET)
               && (CT <= UPPER_BRACKET))
              mark(C,NewCatNumber)
           else if (CT > UPPER_BRACKET)
               a_cat(C)

        T <- #descendents(N)
        if (T >= LOWER_BRACKET)
           mark(N,NewCatNumber)
```

Figure 4.3: Algorithm for creating categories from WordNet's noun hierarchy.

	United States Constitution	*Genesis*
0	assembly (court, legislature)	deity divinity god
1	due_process_of_law	relative relation (mother, aunt)
2	legal_document legal_instrument	worship
3	administrative_unit	man adult_male
4	body (legislative)	professional
5	charge (taxes)	happiness gladness felicity
6	administrator decision_maker	woman adult_female
7	document written_document	evildoing transgression
8	approval (sanction, pass)	literary_composition
9	power powerfulness	religionist religious_person

Figure 4.4: Output using original category set on two well-known texts.

then the procedure is called recursively on the child. Otherwise, the child is too small and is left alone. After all of N's children have been processed, the category that N will participate in has been made as small as the algorithm will allow. There is a chance that N and its unmarked descendants will now make a category that is too small, and if this is the case, N is left alone, and a higher-up node will eventually subsume it (unless N has no parents remaining). Otherwise, N and its remaining unmarked descendants are bundled into a category.

If N has more than one parent, N can end up assigned to the category of any of its parents (or none), depending on which parent was accessed first and how many unmarked children it had at any time, but each synset is assigned to only one category.

The function "mark" places the synset and all its descendents that have not yet been entered into a category into a new category. Note that #descendents is recalculated in the third-to-last line in case any of the children of N have been entered into categories.

In the end there may be isolated small pieces of hierarchy that aren't stored in any category, but this can be fixed by a cleanup pass, if desired.

4.5.2 Assigning Topics using the Original Category Set

Using the 726 categories derived from WordNet, the category assignment algorithm produces the output shown in Figure 4.4 for two well-known texts (made available online by Project Gutenberg). The first column indicates the rank of the category, the second column indicates the score for comparison purposes, and the third column shows the words in the synset at the top-most node of the category (these are not always entirely descriptive, so some glosses are provided in parentheses).

Note that although most of the categories are appropriate (with the glaring exception of "professional" in *Genesis*), there is some redundancy among them, and in some cases they are too fine-grained to indicate main topic information.

In an earlier implementation of this algorithm, the categories were in general larger but less coherent than in the current set. The larger categories resulted in better-trained classifications, but the classes often conflated quite disparate terms. The current implementation produces smaller, more coherent categories. The advantage is that a more distinct meaning can be associated with a particular label, but the disadvantage is that in many cases so few of the words in the category appear in the training data that a weak model is formed. Then the categories with little distinguishing training data dominate the labeling scores inappropriately.

In the category-derivation algorithm described above, in order to increase the size of a given category, terms must be taken from nodes adjacent in the hierarchy (either descendants or siblings). However, adjacent terms are not necessarily closely related semantically, and so after a point, expanding the category via adjacent terms introduces noise. To remedy this problem, WordSpace is used to determine which categories are semantically related to one another, despite the fact that they come from quite different parts of the hierarchy, so they can be combined to form schema-like associations.

4.5.3 Combining Distant Categories

To find which categories should be considered closest to one another, we first determine how close they are in WordSpace (Schütze 1993b) and then group categories together that mutually ranked one another highly.[6] WordSpace is a corpus-based method for inducing semantic representations for a large number of words from lexical coocurrence statistics. The medium of representation is a multi-dimensional, real-valued vector space. The cosine of the angle between two vectors in the space is a continuous measure of their semantic relatedness.

First-degree closeness of two categories c_i and c_j is defined as:

$$D(c_i, c_j) = \frac{1}{2}\frac{1}{|c_i||c_j|}\sum_{\vec{v}\in c_i}\sum_{\vec{w}\in c_j} d(\vec{v}, \vec{w})$$

where d is:

$$d(\vec{v}, \vec{w}) = \sum_i (v_i - w_i)^2$$

The primary rank of category i for category j indicates how closely related i is to j according to first-degree closeness. For instance rank 1 means that i is the closest category to j, and rank 3 means there are only two closer categories to j than i.

We define second-degree closeness from the primary ranks. Secondary ranking is needed because some categories are especially "popular," attracting many other categories to them; the secondary rank enables the popular categories to retain only those categories

[6] All work involving the WordSpace algorithm was done in collaboration with Hinrich Schütze. We are grateful to Robert Wilensky for suggesting collaboration on this problem.

that they mutually rank highly. To determine that close association is mutual between two categories, we check for mutual high ranking. Thus category i and j are grouped together if and only if i ranks j highly and j ranks i highly (where "highly" was determined by a cutoff value – i and j had to be ranked k or above with respect to each other, for a threshold k).

The results of this algorithm are best interpreted via a graphical layout. Figure 4.5 shows a piece of a network created using a presentation tool (Amir 1993) based on theoretical work by Fruchtermann & Rheingold (1990). The underlying algorithm uses a force-directed placement model to layout complex networks (edges are modeled as springs; nodes linked by edges are attracted to each other, but all other pairs of nodes are repelled from one another).

In these networks only connectivity has meaning; distance between nodes does not connote semantic distance. The connectivity of the network is interesting also because it indicates the interconnectivity between categories. From Figure 4.5, we see that categories associated with the notion *sports*, such as *athletic_game*, *race*, *sports_equipment*, and *sports_implement*, have been grouped together. *Athletics* is linked to *vehicle* and *competition* categories; these in turn link to *military_vehicles* and *weaponry* categories, which then lead in to *legal* categories.

The network also shows that categories that are specified to be near one another in WordNet, such as the categories related to *bread*, are found to be closely interrelated. This is useful in case we would like to begin with smaller categories, in order to eliminate some of the large, broad categories that we are currently working with.

Most of the connectivity information suggested by the network was used to create the new categories. However, many of the desirable relationships do not appear in the network, perhaps because of the requirement for highly mutual co-ranking. If we were to relax this assumption we may find better coverage, but perhaps at the cost of more misleading links. The remaining associations were determined by hand, so that the original 726 categories were combined into 106 new *super-categories*.

4.5.4 Revised Topic Assignments

The super-categories are intended to group together related categories in order to eliminate topical redundancy in the labeler and to help eliminate inappropriate labels (since the categories are larger and so have more lexical items serving as evidence). Thus the top four or five super-categories should suffice to indicate the main topics of documents.

Figure 4.6 compares the results of the labeler using the original categories against the super-categories. The numbers beside the category names are the scores assigned by the algorithm; the scores in both cases are roughly similar. It is important to realize that only the top four or five labels are to be used from the super-categories; since each super-category subsumes many categories, only a few super-categories should be expected to contain the most relevant information. The first article is a 31-sentence magazine article, published in 1987, taken from Morris (1988). It describes how Soviet women have little political power, discusses their role as working women, and describes the benefits of college life. The second article is a 77-sentence popular science magazine article about the Magellan space probe exploring Venus. When using the super-categories, the labeler avoids grossly inappropriate labels such as "mollusk_genus" and "goddess" in the Magellan article, and combines categories such as "layer", "natural_depression", and "rock stone" into the one super-category "land terra_firma".

Looking again at the longer texts of the *United States Constitution* and *Genesis* we see in Figure 4.7 that the super-categories are more general and less redundant than the categories shown in Table 4.4. (Although the high scores for the "breads" category seems incorrect, even though the term "bread" occurs 25 times in *Genesis*.) In some cases the user might desire more specific categories; this experiment suggests that the labeler can generate topic labels at multiple levels of granularity.

Section 4.4 evaluates the results of assigning topics based on the supercategories; however we have not rigorously compared the supercategories against the original categories.

4.6 Conclusions

This chapter has presented an algorithm that automatically assigns multiple main topic categories to texts, based on computing the posterior probability of the topic given its surrounding words, without requiring pre-labeled training data or heuristic rules. The algorithm significantly outperforms a baseline measure and approaches the levels of inter-indexer consistency displayed by nonprofessional human indexers. The chapter also describes the construction of a general category set from a hand-built lexical hierarchy. The structure of the WordNet hyponym hierarchy is large and uneven; the bracketing algorithm provides a simple and effective way to automatically subdivide it. The algorithm that uses WordSpace to combine distant parts of the hierarchy is partially effective, but requires a manual postprocessing pass.

The categorization algorithm is effective on texts that have strong thematic discussions, but many kinds of improvements and alternatives remain to be explored. If a document contains terms which are members of small categories, or categories whose terms occur only rarely, then the algorithm erroneously assigns too much weight to these rarer senses. An analysis of the terms whose weights are most strongly associated with each category would be useful for analyzing how to fix this problem. Finally, because the goal of the algorithm is to allow assignment of multiple categories to documents, in the cases in which several categories have significant overlap in meaning, e.g., *reptiles* and *birds*, the algorithm tends to assign both categories to the document, even though a human indexer usually would not.

Fisher (1994) has performed a series of experiments that compare variations of this algorithm. Preliminary results indicate that using direct counts of category membership can improve the results.

It would be interesting to try the training loop idea in which the output of TextTiling is used as input to the category training algorithm, and so on, improving both algorithms simultaneously. This is an area for future work.

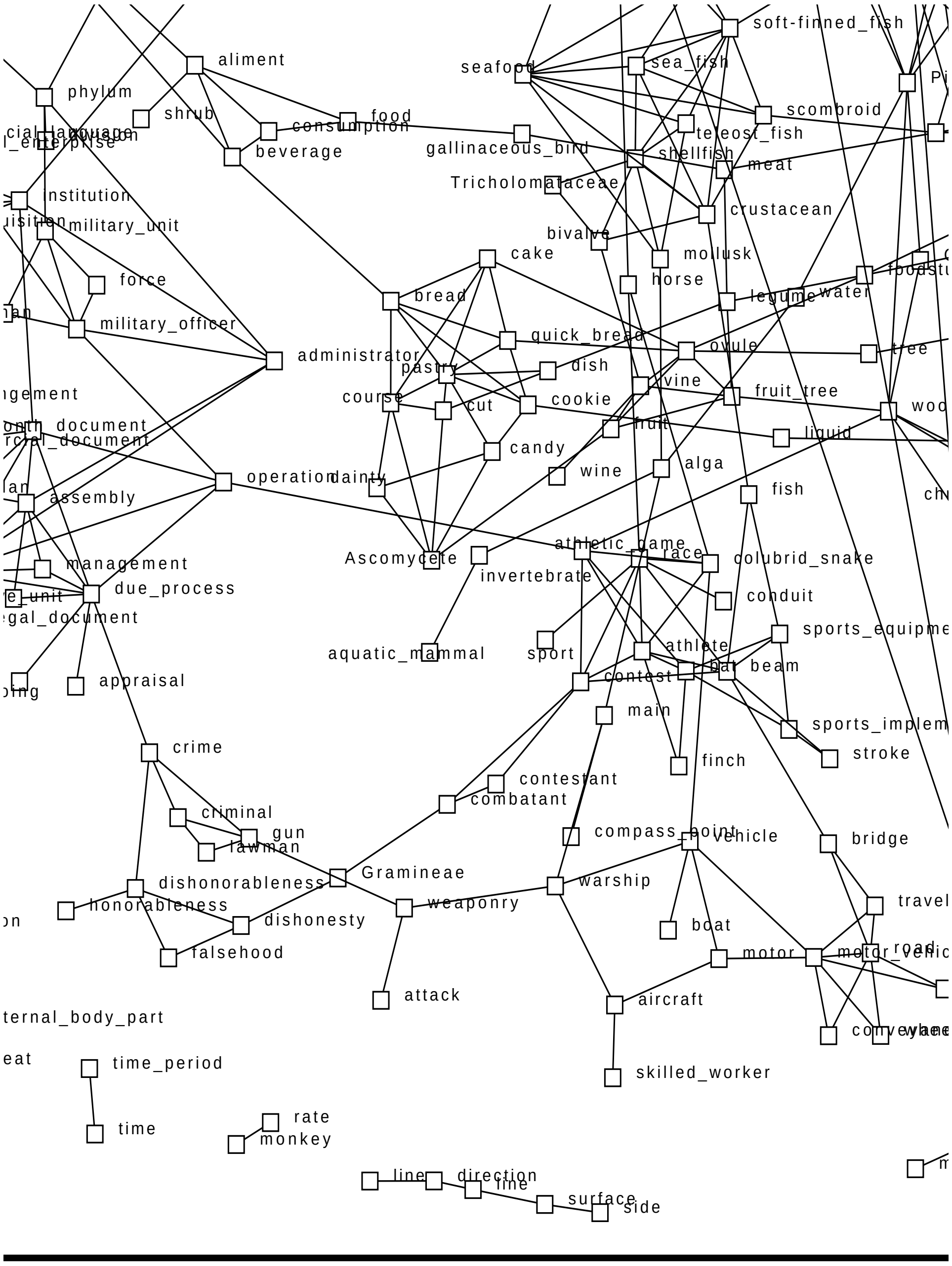


Figure 4.5: A piece of the category network. The grouping algorithm finds relatedness between categories that are near one another in WordNet (e.g., the food terms) as well as categories that are far apart (e.g., "sports equipment" with "athlete").

Raisa Gorbachev article		
	Original Categories	*Super-Categories*
0	woman adult_female	social_standing
1	status social_state	education
2	man adult_male	politics
3	political_orientation ideology	legal_system
4	force personnel	people
5	charge	psychological_state
6	relationship	socializing
7	fear	social_group
8	attitude	personal_relationship
9	educator pedagogue	government

Magellan space probe article		
	Original Categories	*Super-Categories*
0	celestial_body heavenly_body	outer_space
1	mollusk_genus	light_and_energy
2	electromagnetic_radiation	atmosphere
3	layer (surface)	land terra_firma
4	atmospheric_phenomenon	physics
5	physical_phenomenon	arrangement
6	goddess	shapes
7	natural_depression depression	water_and_liquids
8	rock stone	properties
9	space (hole)	amounts

Figure 4.6: Comparison of original and super categories.

United States Constitution		
	Original Categories	*Super-Categories*
0	assembly (court, legislature)	legal_system
1	due_process_of_law	government
2	legal_document legal_instrument	politics
3	administrative_unit	conflict
4	body (legislative)	crime
5	charge (taxes)	finance
6	administrator decision_maker	social_standing
7	document written_document	honesty
8	approval (sanction, pass)	communication

Genesis		
	Original Categories	*Super-Categories*
0	deity divinity god	religion
1	relative relation (mother, aunt)	breads
2	worship	mythology
3	man adult_male	people
4	professional	social_outcasts
5	happiness gladness felicity	social_group
6	woman adult_female	psychological_state
7	evildoing transgression	personality
8	literary_composition	literature

Figure 4.7: Comparison of original and super categories for two well-known texts.

Chapter 5

Multiple Main Topics in Information Access

5.1 Introduction

In this Chapter I address some issues relating to display of results of retrieval from full-text collections. I claim that displaying query results in terms of inter-document similarity is inappropriate with long texts, and suggest instead assigning categories that correspond to documents' main topics. I argue that main topics of long texts should be represented by multiple categories, since in many cases one category cannot adequately classify a text. The display makes use of the automatic categorization algorithm described in Chapter 4. I introduce Cougar, a browsing interface that presents a simple mechanism for displaying multiple category information.

An increasingly important concern to information access is that of passage retrieval from full-text document collections. Full-length expository texts can be thought of in terms of a sequence of subtopical discussions tied together by one or more main topic discussions (see Chapter 3). Two different passages, both of which share terms with a query, may originate in documents with entirely different main topic discussions. For example, Figure 5.1 shows a sketch in which three different passage-level discussions of volcanic activity take place in three different main topic contexts (exploration of Venus, Roman history, and the eruption of Mt. St. Helens). Users should receive some indication of the contexts from which a set of retrieved passages originated in order to decide which passages are worth further scrutiny.

In the text retrieval scenario of retrieval of passages from long texts it is important to supply the user with information that places the results in a meaningful context. Most existing approaches to display of retrieval results can be characterized in two ways: all of the returned documents are displayed either (i) according to their overall similarity to one another, or (ii) in terms of user-selected keywords or attributes they are associated with. I suggest an alternative viewpoint with the following characteristics:

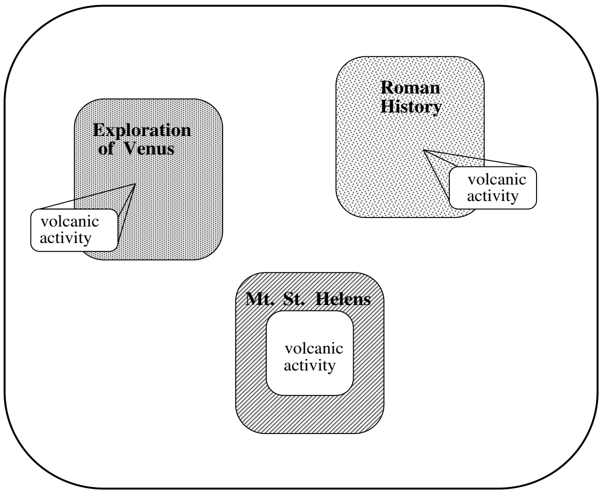


Figure 5.1: Retrieval of passages from full-length text: the contexts in which the localized discussions take place may be entirely different from one another.

- The documents' contents are represented by multiple independent attributes that characterize the main topics of the text.
- The system displays all and only the attributes or topics that are assigned as a result of the query, as opposed to displaying documents that meet pre-selected attributes.
- The system allows display of interactions among the attributes.

The next section expands on the discussion of related work and explains the drawbacks of the two most common retrieval display options with respect to passage retrieval and dataset familiarization. Section 5.3 presents an alternative approach which makes use of category information in order to indicate the main topic discussions of texts. Section 5.4 summarizes the chapter.

5.2 Current Approaches

Textual information does not conform to the expectations of sophisticated display paradigms, such as those seen in the Information Visualizer (Robertson *et al.* 1993). These techniques either require the input to be structured (e.g., hierarchical, for the Cone Tree) or scalar along at least one dimension (e.g., for the Perspective Wall). However, the aspects of a document that satisfy these criteria (e.g., a timeline of document creation dates) do not illuminate the actual content of the documents.

The simplest approach to displaying retrieval results is, of course, to list the titles or first lines of the retrieved documents. One alternative, the TileBar display, is described in Chapter 3. Other systems that do more than this can be characterized as performing one of two functions:

(1) Displaying the retrieved documents according to their overall similarity to other retrieved documents, and/or

(2) Displaying the retrieved documents in terms of keywords or attributes pre-selected by the user.

Both of these approaches, and their drawbacks, are discussed in the subsections that follow.

5.2.1 Overall Similarity Comparison

Several systems display documents in what can be described as a similarity network. A focus document, usually one that the user has expressed interest in, is shown as a node in the center of the display, and documents that are similar to the focus document are represented as nodes linked by edges surrounding the focus document node. Here similarity is measured in terms of the vector space model or a probabilistic model's measure of probability of relevance.

Systems of this kind include the Bead system (Chalmers & Chitson 1992), which displays documents according to their similarity in a two-dimensional rendition of multi-dimensional document space, I^{3}R (Thompson & Croft 1989) and the system of Fowler *et al.* (1991), which display retrieved documents in networks based on interdocument similarity.

A different way to display documents according to their inter-similarity is to cluster the results of the retrieval and make visible the cluster centroids and the distance of the documents from each centroid. Scatter-Gather (Cutting *et al.* 1992), (Cutting *et al.* 1993) is an innovative, query-free browsing technique that allows users to become familiar with the contents of a corpus by interactively clustering subparts of the collection to create table-of-contents-like descriptions. This technique is very effective on shorter texts but, as argued below, will probably be less effective on collections of longer texts. Additionally, Scatter/Gather emphasizes query-free browsing, although it could be augmented with Boolean and similarity search.

Drawbacks of Comparing Full-Length Texts

Most (non-Boolean) information retrieval systems use inter-document similarity to compare documents to a query and determine their relevance. For example, the vector space model of similarity search (Salton 1988), clustering, e.g., (Cutting *et al.* 1992), (Griffiths *et al.* 1986), and latent semantic indexing for determining inter-document similarity, e.g., (Deerwester *et al.* 1990), (Chalmers & Chitson 1992), all work by comparing the entire content of a document against the entire contents of other documents or queries.

These modes of comparison are appropriate on abstracts because most of the (non-stopword) terms in an abstract are salient for retrieval purposes, because they act as place-holders for multiple occurrences of those terms in the original text, and because these terms tend to pertain to the most important topics in the text. When short documents are compared via the vector-space model or clustering, they are positioned in a multi-dimensional space where the closer two documents are to one another, the more topics they are presumed to have in common. This is often reasonable because when comparing abstracts, the goal is to discover which pairs of documents are most alike. For example, a query against a set of medical abstracts which contains terms for the name of a disease, its symptoms, and possible treatments is best matched against an abstract with as similar a constitution as possible.

A problem with applying standard information retrieval methods to full-length text documents is that the structure of full-length documents is quite different from that of abstracts. One way to view an expository text, as mentioned in Chapter 3, is as a sequence of subtopics set against a "backdrop" of one or more main topics. The main topics of a text are discussed in the document's abstract, if one exists, but subtopics usually are not mentioned.

Most long texts discuss several main topics simultaneously; thus, two texts with one shared main topic will often differ in their other main topics. Some topic co-occurrences are more common than others; e.g., terrorism is often discussed in the context of U.S. foreign policy with the Middle East, and these two themes might even be grouped together in some domain-specific ontologies. However, texts often discuss themes that would not usually be considered to be in the same semantic frame; for example, Morris (1988) includes an article that describes terrorist incidents at Bolshoi ballet performances. Therefore, I hypothesize that algorithms that successfully group short texts according to their overall similarity (e.g., clustering algorithms, vector space similarity, and LSI), will produce less meaningful results when applied to full-length texts.

This hypothesis is supported by the fact that recently researchers experimenting with retrieval against datasets consisting of long texts have been breaking the texts into subparts, usually paragraphs, and comparing queries against these isolated pieces (e.g., Salton *et al.* (1993), Salton & Buckley (1992), Al-hawamdeh *et al.* (1991)). These studies find that matching a query against the entirety of a long text is less successful than matching against individual pieces. As further evidence, Voorhees (1985) performed experiments (on standard short-text collections) which found that the cluster hypothesis did not hold; that is, it was not the case that the associations between clustered documents conveyed information about the relevance of documents to requests.

In summary, I claim that when long documents are displayed according to how similar they are throughout, it can be difficult to discern why they were grouped together if this grouping is a function of some intermediate position in multi-dimensional space. If instead we recognize that long texts can be classified according to several different main topics, and contain as well a sequence of subtopical discussions, we have a new basis on which to determine in what ways long documents are similar to one another. This chapter focuses only on accounting for main topic information; the recognition of subtopic structure for information retrieval is a problem unto itself and is discussed in Chapter 3.

5.2.2 User-specified Attributes

Many systems show the relation of the contents of the texts to user-selected attributes; these include VIBE (Korfhage 1991), the InfoCrystal (Spoerri 1993), the Cube of Contents (Arents & Bogaerts 1993), and the system of Aboud *et al.* (1993).

These systems require the users to select which the classifications the display should be organized around. The goal of VIBE (Korfhage 1991) is to display the contents of the entire document collection in a meaningful way. The user defines N "reference points" (which can be weighted terms or term weights) which are placed in various positions in the display, and a document icon is drawn in a location that indicates the distance between the document and all the relevant reference points.

Two interesting graphical approaches are the InfoCrystal and the Cube of Contents. The InfoCrystal (Spoerri 1993) is a sophisticated interface which allows visualization of all possible relations among N attributes. The user specifies which N concepts are of interest (actually Boolean keywords in the implementation, but presumably any kind of labeling information would be appropriate) and the InfoCrystal displays, in an ingenious extension of the Venn-diagram paradigm, the number of documents retrieved that have each possible subset of the N concepts. When the query involves more than four terms the crystals become rather complicated, although there is a provision to build up queries hierarchically. Figure 5.2 shows a sketch of what the InfoCrystal might display as the results of a query against four keywords or Boolean phrases, labeled A, B, C, and D. The diamond in the center indicates that one document was discovered that contains all four keywords. The triangle marked with "12" indicates that twelve documents were found containing attributes A, B, and D, and so on.

The Cube of Contents of (Arents & Bogaerts 1993) is used to help a user build a query by selecting values for up to three mutually exclusive attributes (see Figure 5.3). This assumes a text pre-labeled with relevant information and an understanding of domain-dependent structural information for the document set. Note that this is used to specify the query although it could be used to characterize retrieval results as well. Note that only one intersection of two or three attributes is viewable at any time.

The system of Aboud *et al.* (1993), allows the user to specify multiple class criteria, where the classes are specified in a hierarchy, to help narrow or expand the search set.

The limitations with these approaches are:

(3a) The attributes in question are simply the keywords the user specified in the query, and so do not add information about the contents of the texts retrieved, and/or

(3b) The user must expend effort to choose the attributes to be displayed, and/or

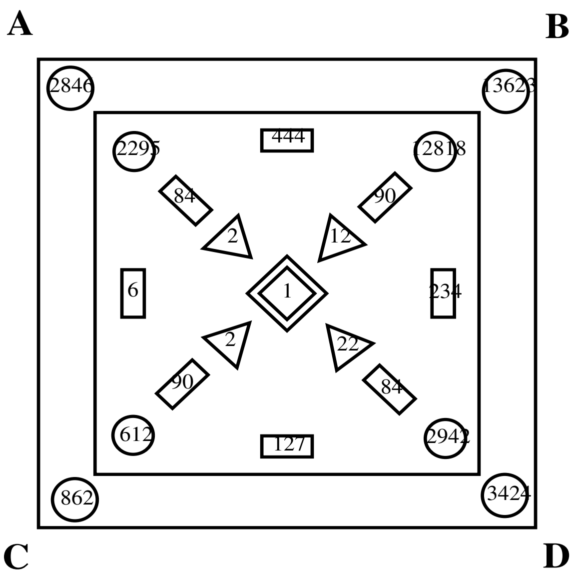


Figure 5.2: The InfoCrystal (Spoerri 1993).

(3c) The user might select attributes that do not correspond to the retrieved documents, thus undercutting the goal of supplying information about the documents returned in response to a general query.

These problems can be easily remedied; the point here is that the standard goal of such systems is to facilitate query construction with attribute information, rather than enhancing display of retrieval results. Note, however, that none of these display paradigms can impart the term distribution information that TileBars do (see Chapter 3).

To summarize this section, previous approaches to displaying retrieval results either display documents in terms of their overall similarity to one another, in terms of similarity to clusters formed from the corpus or the retrieval set, or in terms of attributes preselected by the user. I have discussed problems with each of these approaches. The next section presents an alternative in which these drawbacks are eliminated.

5.3 Multiple Main Topic Display

As mentioned in the Section 5.1, I propose an approach in which multiple independent categories are assigned to the "main topics" of each document[1]. I emphasize the importance of displaying all and only the attributes that are actually assigned to retrieved documents, rather than requiring the user to specify in advance which topics are of interest. This circumvents problems arising from erroneous guesses and reduces the mental effort required

[1]In this discussion, the terms *attribute, topic,* and *category* are interchangeable.

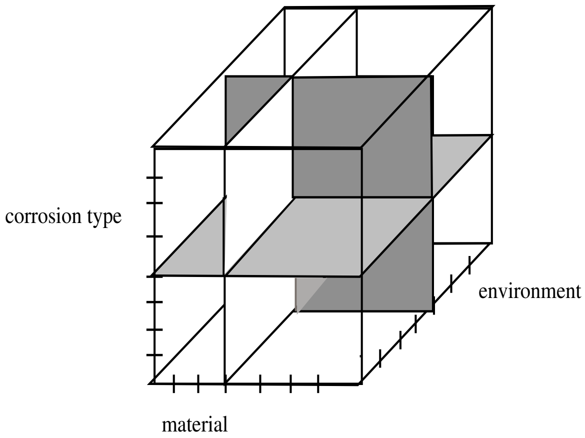


Figure 5.3: The Cube of Contents (Arents & Bogaerts 1993).

by the user when generating initial queries. It also allows for an element of serendipity, both in terms of which categories are displayed and what kinds of interactions among categories may occur. This also prevents clutter resulting from display of attributes that are not present in any retrieved documents[2].

5.3.1 Displaying Frequent Terms

An alternative to assigning documents pre-defined labels is to simply show the documents' most frequent terms. Although top-frequency terms are often very descriptive, problems with using term frequencies arise when the contents of many different documents (or their passages) are displayed simultaneously. One problem is that because there are many different words that contribute to the expression of one concept, it will often be the case that two documents that discuss some of the same main topics will have little overlap in the terms they use to do so. This means that the display will not be able to reveal overlapping themes.

The second problem is that within the display of the most frequent terms for a document, several different terms will contribute to one theme. For example, in a chapter of de Tocqueville (1835), among the most frequent terms are: *judicial, judge, constitution, political, case, court, justice, magistrate* as well as: *American, authority, nation, state.* Thus there is considerable redundancy with respect to what kind of information is being conveyed by the display of the most frequent terms.

[2]Although in domain-specific situations it may be useful to show the user which attributes are missing.

5.3.2 Displaying Main Topic Categories

Instead of or along with frequent term information, category information can indicate the context in which retrieved passages reside. Assigning multiple independent categories allows for recognizing different interactions among documents: two topic categories that are not usually considered semantically similar can nevertheless be associated with the same text if it happens to be about both topics.

If multiple main topic categories are associated with each text, users can browse the results of initial queries with respect to these. Of course, the category sets should be tailored to the text collections they are assigned to. For example, a user interested in local area networks might tap into a general-interest test collection. In this case, when the user queries on the word "LAN", the system returns general categories, i.e. *technology*, *finance*, *legal*, etc. If the user is interested in, say, the impact of LAN technology on the business scene, then this dataset may be useful.

If on the other hand the user wants technical information, the contextualizing information makes it clear that the search should be taken to another dataset. If the same query on a new dataset returns categories like *file servers*, *networks*, *CAD*, etc, then the user can conclude that a technical dataset has been found, and can make subsequent queries more technical in nature.

Library catalog systems have long provided categorization information in the form of subject headings. Researchers have reported that these kinds of headings often mismatch user expectations (Svenonius 1986), (Lancaster 1986). Noreault *et al.* (1981) report on an experiment in which very little overlap occured in search results using controlled vocabulary versus free terms, even though the searches were done by the same professional searcher, in response to the same queries issued against the same dataset. However, there is also evidence that when such subject heading information is combined with free text search, results are improved (Markey *et al.* 1982), (Henzler 1978), (Lancaster 1986). Here I am suggesting the combination of category information with term search capabilities.

5.3.3 A Browsing Interface

Because several categories can be associated with each retrieved document, a method for browsing this multi-dimensional space is needed. One approach to the display of multi-dimensional information is to provide the user with a simple way to control which attributes are seen at each point in time. The interface described here allows users to view the results of the query graphically, according to the intersection of assigned categories, using a Venn diagram paradigm.[3] The interface, called Cougar, combines keyword and category information – users can search on either kind of information or both (see Figure 5.4). This allows users to get a feeling for document similarity based on the main topic categories they

[3] Michard (1982) uses a Venn diagram in a study about its effectiveness in helping novice users create Boolean queries, using the graphical notion of intersection to indicate conjunction of terms. The diagram is not used for display of results or for conjoining more than three terms.

share. Note that different documents can be grouped together as being similar based on which categories are being looked at. E.g., if one document is about the cost of removing contaminants from food and another the cost of removing contaminants from an ecological disaster, when viewed according to the *finance* category they have an intersection, whereas if the *finance* category is not selected, the two documents do not appear to have similarities.

This particular cut on how to display information begins with a fixed set of categories, membership in which is designed to correspond to users' intuitions. Of course this approach is flawed, both because no one set of category choices is going to fit every document set and because users will have to guess what categorization according to the topic really means. Nevertheless, I posit that this approach is better than requiring the user to guess why a group of long documents have been labeled as being similar to one another, and better than simply looking at a list of titles ranked by vector-space based similarity to the query. Furthermore, since users do not have to specify in advance which categories are of interest, they are less likely to miss interesting documents just because their understanding of the classification procedure is inaccurate.

In Cougar, documents are assigned a fixed number of categories from a pre-determined set using the automatic categorization algorithm described in Chapter 4. In the current system each document is assigned its three top-scoring categories. The documents are then indexed on the category information as well as on all (non-stopword) lexical items from the title and the body. Indexing and retrieval is currently done using Cornell's Smart system (Salton 1971), although this will soon change to the indexing structure used in Chapter 3 (Section 3.4.3). The interface was created using Tcl/Tk (Ousterhout 1991).

Two datasets have been assigned categories and indexed. The first is a subset of a collection of AP news articles taken from month of 1989 from the TIPSTER collection (Harman 1993) and is indexed with the general category set described in Chapter 4. The second is a collection of computer science technical reports, part of the CNRI CS-TR project collection, and is indexed with computer-related categories.

Users issue queries by entering words or selecting categories from an available list. As mentioned above, typically the user only enters term information. After the user initiates the search a list of titles of the top-scoring documents appears. The number of titles displayed is a parameter that is set in Smart; currently 50 documents are retrieved at a time. The top three categories for each document are also retrieved and the most frequently occurring of these are displayed in a bank of color-coded buttons above a Venn diagram skeleton. The user selects up to three of the categories and sees how the documents intersect with respect to those categories. One category can be unselected in order to allow the selection of another; the display of documents in the Venn diagram changes accordingly.

More specifically, the user selects one of the categories by mouse-clicking on a category box. The system paints one of the Venn-diagram rings with the corresponding color and places document ID numbers that have been assigned this category into the part of the ring that indicates no intersection with other categories. Clicking on an ID number causes the corresponding title to be highlighted, and double-clicking brings up a window containing the document itself. The user can now unselect this category, causing the ring to become

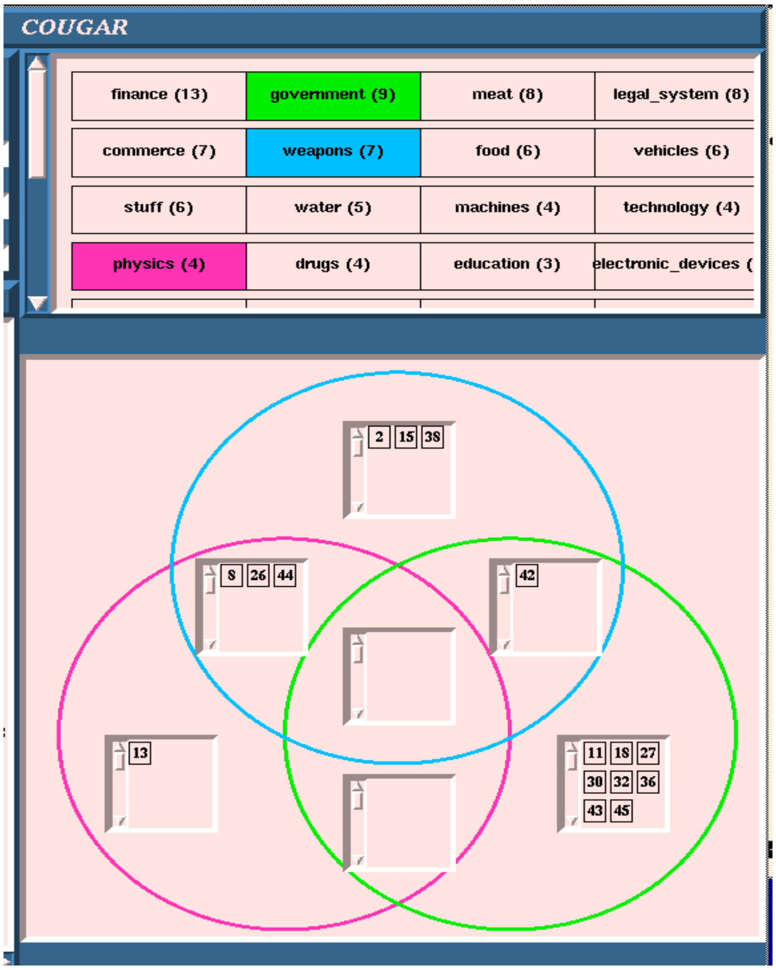


Figure 5.4: The Cougar interface.

uncolored and the displayed document IDs to disappear. Alternatively, the user can choose an additional category, causing an additional ring to be painted and filled in with document IDs. If any of the retrieved documents have been assigned both of the selected categories, their ID numbers are displayed in the appropriate intersection region. Once all three rings have been assigned categories, the user must unselect one category before selecting a new one. In this way users can easily vary which subset of the category sets is active.

Keywords in Context

Figure 5.4 shows a configuration in which all three categories have been selected. Bearing in mind that the documents retrieved are ones in which the term *contaminant* appears, we can examine the kind of context provided by the category information. The most frequently assigned categories include *finance, government, meat, legal_system, commerce, weapons, food,* and *vehicles*. As the categories imply, discussions of contaminants occur in many different contexts.

Document 42, at the intersection of government and weapons, discusses a government proposal to cleanup a nuclear weapons production complex. Documents 8, 26, and 44, at the intersection of physics and weapons, discuss the reopening of a plutonium processing plant, obstacles to the development of orbiting nuclear reactors, and modernization of nuclear reactors. Document 13 describes a nuclear waste leak, document 38 the risks of the launch of a satellite containing plutonium, and document 30 discusses the Reagan administration's record in treating the ozone layer.

One article labeled with *ships, bodies_of_water,* and *nature* describes the effects of an oil spill on birdlife. Articles labeled with the *food* category include two about an incident of cyanide poisoning in yogurt. Note that if a user were interested in documents that talk about contamination in food, in order to discover this article using keywords alone, the user would have had to specify all food terms of interest. However, with appropriate category information this is not necessary.

Categories to Determine Relevance of Keywords

In the next example, only eight of the top fifty retrieved documents in response to a query on the word *cattle* are labeled with the higher-level category that corresponds to cattle (*herd_animals*). Most of those that are not labeled with *herd_animals* are about financial matters relating to crops and foods (e.g., crop futures). Two of those that are labeled with *herd_animals*, when intersected with *meat* describe cattle in the role of livestock, the third describes a cattle drive, and the fourth, whose other category labels are *countries* and *bodies_of_water*, has only a passing reference to cattle and really describes a murder related to land ownership of tropical rainforests.

By contrast, retrieving on the word *cow* results in articles about land disputes with Native Americans (at the intersection of *government, herd_animals*, and *legal_system*) and grazing fees. One document that is not labeled with *herd_animals* but instead with *crime, weapons*, and *defense*, has only a passing reference to cows and is about a robbery.

Thus the categories can be used to show whether or not a search term is actually well-represented in a text. If the text is not assigned the category that the search time is a member of, then this is a strong indicator that the term is only discussed in passing.

The set of 106 general categories used to characterize the AP data was derived from WordNet (Miller *et al.* 1990) as described in Chapter 4. The algorithm has also been trained on a collection of computer science technical reports using a set of 11 categories derived from a loose interpretation of the ACM Computing Reviews classifications.

5.3.4 Discussion

The AIR/SCALIR system (Rose & Belew 1991) has an interface that most closely incorporates the goals set forth here. The system allows for very simple queries, and provides a kind of contextualizing information. A connectionist network determines in advance a set of terms that characterize documents from a collection of bibliographic records. When the user issues a query, the system retrieves documents that contain the terms of the query (restricting the number of documents that are displayed at any one time). Additional terms that are strongly associated with the retrieved documents are also retrieved. The system displays three rows of nodes corresponding to the associated terms, the documents, and the authors of the documents, respectively. The term nodes are connected to the document nodes via edge links, so the user can see which documents are associated with each important term. Only those terms relevant to the retrieved documents are shown, although the documents retrieved are influenced to some extent by which associated terms are retrieved. Figure 5.6 is a sketch of the interface's output when presented with the query ((:TERM "ASSOCIATIVE")(:AUTH "ANDERSON, J.A.")).

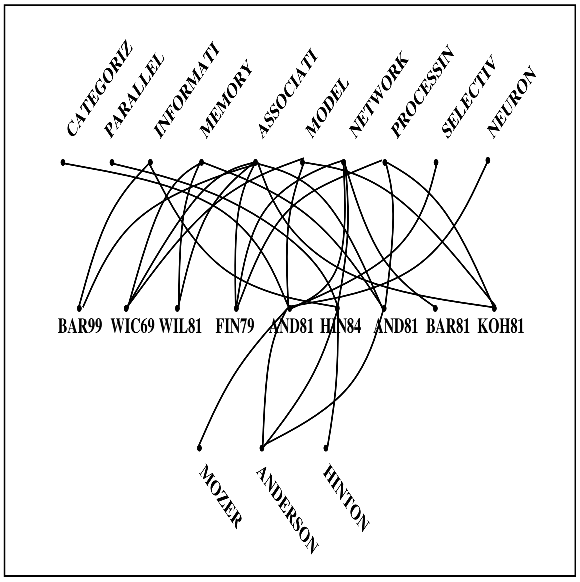


Figure 5.6: A sketch of the AIR system interface (Rose & Belew 1991).

The AIR interface differs from that suggested here in that it is not geared toward displaying subsets of interacting attributes. For this reason, it appears that if there are a large number of links between associated terms and documents, or if the links are not neatly organized, the relationships will be difficult to discern. Furthermore, categorizing information is not geared toward characterizing full-text documents. However, the approach presented here might benefit by incorporating an option to display the categories and documents in a similar manner.

Similarly, rather than using a Venn diagram display, the four-attribute InfoCrystal (Spoerri 1993) might be a useful alternative, applied as suggested here to display subsets of the relevant categories.

5.4 Conclusions

A full-fledged information access system should consist of several different tools for query formulation, document indexing, dataset selection, and characterization of retrieval results. I have described an information access situation – passage retrieval from full-length texts – in which users can benefit from an interface that displays information about the main topic contexts of the retrieved documents.

Users requesting passages from long texts do not know the contexts from which the passages were extracted. Existing approaches either (i) show how similar documents are to one another or the query, or (ii) require users to specify terms or attributes to organize the resulting documents around. I have described problems with both approaches and suggested that retrieval results be displayed in terms of multiple independent attributes that characterize the main topics of the texts, and that the system volunteer display of the relevant attributes, rather than require the user to guess them.

The attributes or categories can vary depending on what kind of information is available and/or appropriate for the corpus. I have suggested assigning categories that characterize the main topics of long texts, and have described an algorithm that can do so with some degree of success without requiring pre-labeled texts. I anticipate improvement in automated category assignment algorithms in future.

A consequence of allowing multiple categories to be assigned to documents is that they make the display problem a multi-dimensional one. To handle this, I suggest a mechanism that gives the user some control over which categories are at the focus of attention at any given time, and a simple way to see how the retrieved documents are related to one another with respect to these categories.

I have implemented a prototype of this display paradigm; it illustrates the main points behind the ideas presented here although user evaluation studies remain to be done. In future I plan to incorporate these mechanisms into an interface for for querying against subtopic structure, and for allowing queries to specify subtopic terms with respect to main topic categories, like that described in Section 3.

Although I have not formally evaluated the Cougar display, anecdotal user reaction is positive. Users find appealing the ability to switch among the category assignments and see the resulting topic intersections. When the topic assignments are incorrect, however, the tool is probably worse than no tool at all. Furthermore, the highest-ranked categories for a document are those that are similar in meaning. This detracts from the goal of showing the interaction of the disparate main topics. Thus an improved category assignment algorithm should improve the appeal of the tool.

Chapter 6

Conclusions and Future Work

In this dissertation I have introduced new ways to view and analyze the structure of full-text documents. I have investigated the role of contextual information in the automated retrieval and display of full-text documents, using computational linguistics algorithms to automatically detect structure in and assign topic labels to texts. I have shown how, for the purposes of information access, full texts are qualitatively different from abstracts, and have suggested that as a consequence, full text requires new approaches to information access. As a first step, I have suggested the examination of patterns of term distribution in long texts, and have shown how these patterns are useful both for recognizing subtopic structure and for describing the results of a query. I have also argued, following Cutting *et al.* (1990), that the mechanisms for querying and displaying documents should receive as much attention as the retrieval algorithm, and that all three components should mutually reinforce one another.

I have described an algorithm, called TextTiling, that uses lexical frequency and distribution to identify the subtopic structure of expository texts. The currently most successful version of this algorithm requires only a short stoplist and a morphological analyzer and analyzes about 20 megabytes of text an hour (including tokenization). I have also described an algorithm that assigns multiple main topic categories to long texts. This algorithm requires a pre-defined category set and a training run but does not require pre-labeled texts, which are much harder to come by than category sets. Both algorithms are compared against reader judgments and are found to perform well, although not flawlessly, on these approximate tasks.

I have also presented a framework for the interpretation of query term distribution patterns within full text documents. This analysis leads to a new interface paradigm, called TileBars, that provides a compact and informative iconic representation of the documents' contents with respect to the query terms. TileBars allow users to make informed decisions about not only which documents to view, but also which passages of those documents, based on the distributional behavior of the query terms in the documents. I have demonstrated the use of TileBars in an analysis of some of the TREC queries (Harman 1993). In the course of this analysis I showed that the patterns of term distributions for relevant documents can vary depending on the query. This supports my conjecture that the criteria upon which ranking is based should be shown explicitly in such a way that users can make informed decisions about which documents to view. It also lends partial support to my hypotheses about the meanings of various patterns of term distribution.

I have also described a new interface, called Cougar, that allows users to view retrieved documents according to multiple main topic assignments. Cougar has an appealing interactive component that allows users to see the main topic context in which retrieved documents are used, providing them with information that can be missing in the TileBar display. Both display tools need to be integrated into one interface, along with other text analysis facilities to allow fast assimilation of the contents of retrieved information. Together, the representation of main topic and subtopic structural information provides a powerful new paradigm for interpreting the results of queries against full-text collections.

The TextTiling work introduces a new granularity of analysis for the text segmentation task. Evidence for the usefulness of multi-paragraph segments is increasing, despite the fact that this is a nontraditional discourse unit. Aside from the use in information access described here, other potential applications of multi-paragraph segments are multiple-window text displays and text window identification for corpus-based natural language processing algorithms.

This work should be extended in several directions. First, I plan to more formally test some of the embellishments to the TextTiling algorithm. I am particularly interested in trying different ways to integrate thesaural terms into the algorithm (perhaps simply by using a good online thesaurus, should one become generally available). I would also like to improve the results in the cases in which the evidence for one paragraph boundary over another is weak (e.g., when a valley in the similarity score plot falls within a paragraph, or when two short paragraphs are adjacent to one another near a valley). One approach is to try simple discourse cues; another is to make a more localized analysis of term overlap when the boundary choice is unclear. Still another alternative is to find a way to express tile overlap, especially when transitions takes place mid-paragraph.

I would also like to formally compare TextTiles to paragraphs in some task. If tiles perform better, or for that matter, no worse than paragraphs in information access tasks, then tiles are preferable for the simple reason that they are less expensive to store and process simply because there are fewer tiles than paragraphs per document (if positional information within tiles or paragraphs is not important).

I would also like to formally evaluate TileBars in terms of their use in relevance feedback and with respect to how users interpret the meaning of the term distributions. The analysis could compare user's expectations about the meaning of the term distributions against the analysis shown in the chart of Chapter 3. It may be useful to determine in what situations the users' expectations are not met, in hopes of identifying what additional information should be added in order to prevent misconceptions.

Both display mechanisms described here should be extended to work with texts that already do have some hierarchical structure built in.

The information access community needs to develop a passage retrieval test collection.

The examples and discussion in this dissertation suggest that such a collection should be cognizant of issues relating to term distribution: relevance judgments should indicate what topical or distributional role the query terms are to play within the retrieved documents.

Appendix A

Tocqueville, Chapter 1

The text of *Democracy in America*, by Alexis de Tocqueville, 1835, Volume 1 Chapter 1.

DOC SEGMENT 1

1 North America presents in its external form certain general features which it is easy to distinguish at the first glance.

2 A sort of methodical order seems to have regulated the separation of land and water, mountains and valleys. A simple but grand arrangement is discoverable amid the confusion of objects and the prodigious variety of scenes.

3 This continent is almost equally divided into two vast regions. One is bounded on the north by the Arctic Pole, and on the east and west by the two great oceans. It stretches towards the south, forming a triangle, whose irregular sides meet at length above the great lakes of Canada. The second region begins where the other terminates, and includes all the remainder of the continent. The one slopes gently towards the Pole, the other towards the Equator.

4 The territory included in the first region descends towards the north with a slope so imperceptible that it may almost be said to form a plain. Within the bounds of this immense level tract there are neither high mountains nor deep valleys. Streams meander through it irregularly; great rivers intertwine, separate, and meet again, spread into vast marshes, losing all trace of their channels in the labyrinth of waters they have themselves created, and thus at length, after innumerable windings, fall into the Polar seas. The great lakes which bound this first region are not walled in, like most of those in the Old World, between hills and rocks. Their banks are flat and rise but a few feet above the level of their waters, each thus forming a vast bowl filled to the brim. The slightest change in the structure of the globe would cause their waters to rush either towards the Pole or to the tropical seas.

DOC SEGMENT 2

5 The second region has a more broken surface and is better suited for the habitation of man. Two long chains of mountains divide it, from one to the other: one, named the Allegheny, follows the direction of the shore of the Atlantic Ocean; the other is parallel with the Pacific.

6 The space that lies between these two chains of mountains contains 228,843 square leagues. Its surface is therefore about six times as great as that of France.

7 This vast territory, however, forms a single valley, one side of which descends from the rounded summits of the Alleghenies, while the other rises in an uninterrupted course to the tops of the Rocky Mountains. At the bottom of the valley flows an immense river, into which you can see, flowing from all directions, the waters that come down from the mountains. In memory of their native land, the French formerly called this river the St. Louis. The Indians, in their pompous language, have named it the Father of Waters, or the Mississippi.

DOC SEGMENT 3

8 The Mississippi takes its source at the boundary of the two great regions of which I have spoken, not far from the highest point of the plateau that separates them. Near the same spot rises another river, which empties into the Polar seas. The course of the Mississippi is at first uncertain: it winds several times towards the north, whence it rose, and only at length, after having been delayed in lakes and marshes, does it assume its definite direction and flow slowly onward to the south.

9 Sometimes quietly gliding along the chalky bed that nature has assigned to it, sometimes swollen by freshets, the Mississippi waters over 1,032 leagues in its course. At the distance of 600 leagues from its mouth this river attains an average depth of 15 feet; and it is navigated by vessels of 300 tons for a course of nearly 200 leagues. One counts, among the tributaries of the Mississippi, one river of 1,300 leagues, one of 900, one of 600, one of 500, four of 200, not to speak of a countless multitide of small trams that rush from all directions to mingle in its flow.

DOC SEGMENT 4

10 The valley which is watered by the Mississippi seems to have been created for it alone, and there, like a god of antiquity, the river dispenses both good and evil. Near the stream nature displays an inexhaustible fertility; the farther you get from its banks, the more sparse the vegetation, the poorer the soil, and everything weakens or dies. Nowhere have the great convulsions of the globe left more evident traces than in the valley of the Mississippi. The whole aspect of the country shows the powerful effects of water, both by its fertility and by its barrenness. The waters of the primeval ocean accumulated enormous beds of vegetable mold in the valley, which they leveled as they retired. Upon the right bank of the river are found immense plains, as smooth as if the tiller had passed over them with his roller. As you approach the mountains, the soil becomes more and more unequal and sterile; the ground is, as it were, pierced in a thousand places by primitive rocks, which appear like the bones of a skeleton whose flesh has been consumed by time. The surface of the earth is covered with a granitic sand and irregular masses of stone, among which a few plants force their growth and give the appearance of a green field covered with the ruins of a vast edifice. These stones and this sand disclose, on examination, a perfect analogy with those that compose the arid and broken summits of the Rocky Mountains. The flood of waters which washed the soil to the bottom of the valley afterwards carried away portions of the rocks themselves; and these, dashed

and bruised against the neighboring cliffs, were left scattered like wrecks at their feet.

11 The valley of the Mississippi is, on the whole, the most magnificent dwelling-place prepared by God for man's abode; and yet it may be said that at present it is but a mighty desert.

12 On the eastern side of the Alleghenies, between the base of these mountains and the Atlantic Ocean, lies a long ridge of rocks and sand, which the sea appears to have left behind as it retired. The average breadth of this territory does not exceed 48 leagues; but it is about 300 leagues in length. This part of the American continent has a soil that offers every obstacle to the husbandman, and its vegetation is scanty and unvaried.

13 Upon this inhospitable coast the first united efforts of human industry were made. This tongue of arid land was the cradle of those English colonies which were destined one day to become the United States of America. The center of power still remains here; while to the west of it the true elements of the great people to whom the future control of the continent belongs are gathering together almost in secrecy.

DOC SEGMENT 5

14 When Europeans first landed on the shores of the West Indies, and afterwards on the coast of South America, they thought themselves transported into those fabulous regions of which poets had sung. The sea sparkled with phosphoric light, and the extraodinary transparency of its waters disclosed to the view of the navigator all the depths of the ocean. Here and there appeared little islands perfumed with odoriferous plants, and resembling baskets of flowers floating on the tranquil surface of the ocean. Every object that met the sight in this enchanting region seemed prepared to satisfy the wants or contribute to the pleasures of man. Almost all the trees were loaded with nourishing fruits, and those which were useless as food delighted the eye by the brilliance and variety of their colors. In groves of fragrant lemon trees, wild figs, flowering myrtles, acacias, and oleanders, which were hung with festoons of various climbing plants, covered with flowers, a multitude of birds unknown in Europe displayed their bright plumange, glittering with purple and azure, and mingled their warbling with the harmony of a world teeming with life and motion.

15 Underneath this brilliant exterior death was concealed. But this fact was not then known, and the air of these climates had an indefinable enervating influence, which made man cling to the present, heedless of the future.

16 North America appeared under a very different aspect: there everything was grave, serious, and solemn; it seemed created to be the domain of intelligence, as the South was that of sensual delight. A turbulent and foggy ocean washed its shores. It was girt round by a belt of granitic rocks or by wide tracts of sand. The foliage of its woods was dark and gloomy, for they were composed of firs, larches, evergreen oaks, wild olive trees, and laurels.

DOC SEGMENT 6

17 Beyond this outer belt lay the thick shades of the central forests, where the largest trees which are produced in the two hemispheres grow side by side. The plane, the catalpa, the sugar maple, and the Virginian poplar mingled their branches with those of the oak, the beech, and the lime.

18 In these, as in the forests of the Old World, destruction was perpetually going on. The ruins of vegetation were heaped upon one another; but there was no laboring hand to remove them, and their decay was not rapid enough to make room for the continual work of reproduction. Climbing plants, grasses, and other herbs forced their way throught the mass of dying trees; they crept along their bending trunks, found nourishment in their dusty cavities, and a passage beneath the lifeless bark. Thus decay gave its assistance to life, and their respective productions were mingled together. The depths of these forests were gloomy and obscure, and a thousand rivulets, undirected in their course by human industry, preserved in them a constant moisture. It was rare to meet with flowers, wild fruits, or birds beneath their shades. The fall of a tree overthrown by age, the rushing torrent of a cataract, the lowing of the buffalo, and the howling of the wind were the only sounds that broke the silence of nature.

19 To the east of the great river the woods almost disappeared; in their stead were seen prairies of immense extent. Whether Nature in her infinite variety had denied the germs of trees to these fertile plains, or whether they had once been covered with forests, subsequently destroyed by the hand of man, is a question which neither tradition nor scientific research has been able to answer.

DOC SEGMENT 7

20 These immense deserts were not, however, wholly untenanted by men. some wandering tribes has been for ages scattered among the forest shades or on the green pastures of the prairie. From the mouth of the St. Lawrence to the Delta of the Mississippi, and from the Atlantic to the Pacific Ocean, these savages possessed certain points of resemblance that bore witness to their common origin; but at the same time they differed from all other known races of men; they were neither white like the Europeans, nor yellow like most of the Asiatics, nor black like the Negroes. Their skin was reddish brown, their hair long and shining, the lips thin, and their cheekbones very prominent. The languages spoken by the North American tribes had different vocabularies, but all obeyed the same rules of grammar. These rules differed in several points from such as had been observed to govern the origin of language. The idiom of the Americans seemed to be the product of new combinations, and bespoke an effort of the understanding of which the Indians of our days would be incapable.

DOC SEGMENT 8

21 The social state of these tribes differed also in many respects from all that was seen in the Old World. They seem to have multiplied freely in the midst of their deserts, without coming in contact with other races more civilized than their own. Accordingly, they exhibited none of those indistinct, incoherent notions of right and wrong, none of that deep corruption of manners, which is usually joined with ignorance and rudeness among nations who, after advancing to civilization, have relapsed into a state of barbarism. The Indian was indebted to no one but himself; his virtues, his vices, and his prejudices were his own work; he had grown up in the wild independence of his nature.

DOC SEGMENT 9

22 If in polished countries the lowest of the people are rude and uncivil, it is not merely because they are poor and ignorant, but because, being so, they are in daily contat with rich and enlightened men. The sight of their own hard lot and their weakness, which is daily contrasted with the happiness and power of some of their fellow creatures, excites in their hearts at the same time the sentiments of anger and of fear: the consciousness of their inferiority and their dependence irritates while it humiliates them. This state of mind displays itself in their manners and language; they are at once insolent and servile. The truth of this is easily proved by observation: the people are more rude in aristocratic counties than elsewhere; in opulent cities than in rural districts. In those places where the rich and powerful are assembled together, the weak and the indigent feel themselves oppressed by their inferior condition. Unable to perceive a single chance of regaining their equality, they give up to despair and allow themselves to fall below the dignity of human nature.

23 This unfortunate effect of the disparity of conditions is not observable in savage life: the Indians, although they are ignorant and poor, are equal and free.

24 When Europeans first came among them, the natives of North America were ignorant of the value of riches, and indifferent to the enjoyments that civilized man procures for himself by their means. Nevertheless there was nothing coarse in their demeanor; they practiced habitual reserve and a kind of aristocratic politeness.

25 Mild and hospitable when at peace, though merciless in war beyond any known degree of human ferocity, the Indian would expose himself to die of hunger in order to succor the stranger who asked admittance by night at the door of his hut; yet he could tear in pieces with this hands the still quivering limbs of his prisoner. The famous republics of antiquity never gave examples of more unshaken courage, more haughty spirit, or more intractable love of independence than were hidden in former times among the wild forests of the New World. The Europeans produced no great impression when they landed upon the shores of North America; their presence engendered neither envy nor fear. What influence could they possess over such men as I have described? The Indian could live without wants, suffer without complaint, and pour out his death-song at the stake. Like all the other members of the great human family, these savages believed in the existence of a better world, and adored, under different names, God, the Creator, of the universe. Their notions on the great intellectual truths were in general simple and philosophical.

DOC SEGMENT 10

26 Although we have here traced the character of a primitive people, yet it cannot be doubted that another people, more civilized and more advanced in all respects, had preceded it in the same regions.

27 An obscure tradition which prevailed among the Indians on the borders of the Atlantic informs us that these very tribes formerly dwelt on the west side of the Mississippi. Along the banks of the Ohio, and throughout the central valley, there are frequently found, at this day, tumuli raised by the hands of men. On exploring these heaps of earth to their center, it is usual to meet with human bones, strange instruments, arms and utensils of all kinds, made of metal, and destined for purposes unknown to the present race. The Indians of our time are unable to give any information relative to the history of this unknown people. Neither did those who lived three hundred years ago, when America was first discovered, leave any accounts from which even a hypothesis could be formed. Traditions, those perishable yet ever recurrent monuments of the primitive world, do not provide any light. There, however, thousands of our fellow men have lived; one cannot doubt that. When did they go there, what was their origin, their destiny, their history? When and how did they disappear? No one can possibly tell.

28 How strange it appears that nations have existed and afterwards so completely disappeared from the earth that the memory even of their names is effaced! Their languages are lost; their glory is vanished like a sound without an echo; though perhaps there is not one which has not left behind it some tomb in memory of its passage. Thus the most durable monument of human labor is that which recalls the wretchedness and nothingness of man.

DOC SEGMENT 11

29 Although the vast country that I have been describing was inhabited by many indigenous tribes, it may justly be said, at the time of its discovery by Europeans, to have formed one great desert. The Indians occupied without possessing it. It is by agricultural labor that man appropriates the soil, and the early inhabitants of North America lived by the produce of the chase. Their implacable prejudices, their uncontrolled passions, their vices, and still more, perhaps, their savage virtues, consigned them to inevitable destruction. The ruin of these tribes began from the day when Europeans landed on their shores; it has proceeded ever since, and we are now witnessing its completion. They seem to have been placed by Providence amid the riches of the New World only to enjoy them for a season; they were there merely to wait till others came. Those coasts, so admirably adapted for commerce and industry; those wide and deep rivers; that inexhaustible valley of the Mississippi; the whole continent, in short, seemed prepared to be the abode of a great nation yet unborn.

30 In that land the great experiment of the attempt to construct society upon a new basis was to be made by civilized man; and it was there, for the first time, that theories hitherto unknown, or deemed impracticable, were to exhibit a spectacle for which the world had not been prepared by the history of the past.

Bibliography

ABOUD, M., C. CHRISMENT, R. RAZOUK, & F. SEDES. 1993. Querying a hypertext information retrieval system by the use of classification. *Information Processing and Management* 29.387–396.

AL-HAWAMDEH, S., R. DEVERE, G. SMITH, & P. WILLETT. 1991. Using nearest-neighbor searching techniques to access full-text documents. *Online Review* 15.173–191.

ALSHAWI, HIYAN. 1987. Processing dictionary definitions with phrasal pattern hierarchies. *American Journal of Computational Linguistics* 13.195–202.

ALTERMAN, RICHARD, & LARRY A. BOOKMAN. 1990. Some computational experiments in summarization. *Discourse Processes* 13.143–174.

AMIR, ELAN, 1993. Carta: A network topology presentation tool. Project Report, UC Berkeley.

ARENTS, H. C., & W. F. L. BOGAERTS. 1993. Concept-based retrieval of hypermedia information – from term indexing to semantic hyperindexing. *Information Processing and Management* 29.373–386.

BACHENKO, JOAN, EILEEN FITZPATRICK, & C.E. WRIGHT. 1986. The contribution of parsing to prosodic phrasing in an experimental text-to-speech system. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, 145–155.

BAREISS, RAY. 1989. *Exemplar-based knowledge acquisition*. Perspectives in Artificial Intelligence. Academic Press, Inc.

BATALI, JOHN, 1991. *Automatic acquisition and use of some of the knowledge in physics texts*. Massachusetts Institute of Technology, Artificial Intelligence Laboratory dissertation.

BATES, MARCIA J. 1986. Subject access in online catalogs a design model. *Journal of the American Society for Information Science* 37.

BELL, JOHN E., & LAWRENCE A. ROWE. 1990. Human factors evaluation of a textual, graphical, and natural language query interfaces. Technical Report M90/12, UC Berkeley ERL.

BERTIN, JACQUES. 1983. *Semiology of graphics*. Madison, WI: The University of Wisconsin Press. Translated by William J. Berg.

BRENT, MICHAEL R. 1991. Automatic acquisition of subcategorization frames from untagged, free-text corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*.

BROWN, GILLIAN, & GEORGE YULE. 1983. *Discourse analysis*. Cambridge Textbooks in Linguistics Series. Cambridge University Press.

BUCKLEY, CHRIS, JAMES ALLAN, & GERARD SALTON. 1994. Automatic routing and ad-hoc retrieval using SMART: TREC 2. In *Proceedings of TREC-2*, ed. by Donna Harman. To appear.

CALZOLARI, NICOLETTA, & REMO BINDI. 1990. Acquisition of lexical information from a large textual italian corpus. In *Proceedings of the Thirteenth International Conference on Computational Linguistics*, Helsinki.

CARDIE, CLAIRE. 1992. Corpus-based acquisition of relative pronoun disambiguation heurisitics. In *Proceedings of the 30th Meeting of the Association for Computational Linguistics*, 216–223.

CHAFE, WALLACE L. 1979. The flow of thought and the flow of language. In *Syntax and semantics: Discourse and syntax*, ed. by Talmy Givón, volume 12, 159–182. Academic Press.

CHALMERS, MATTHEW, & PAUL CHITSON. 1992. Bead: Exploration in information visualization. In *Proceedings of the 15th Annual International ACM/SIGIR Conference*, 330–337, Copenhagen, Denmark.

CHARNIAK, EUGENE. 1983. Passing markers: a theory of contextual influence in language comprehension. *Cognitive Science* 7.171–190.

CHEN, FRANCINE R., & MARGARET WITHGOTT. 1992. The use of emphasis to automatically summarize a spoken discourse. In *Proceedings of ICASSP*.

CHURCH, KENNETH W., & PATRICK HANKS. 1989. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, 76–83.

——, & PATRICK HANKS. 1990. Word association norms, mutual information, and lexicography. *American Journal of Computational Linguistics* 16.22–29.

——, & MARK Y. LIBERMAN. 1991. A status report on the ACL/DCI. In *The Proceedings of the 7th Annual Conference of the UW Centre for the New OED and Text Research: Using Corpora*, 84–91, Oxford.

——, & ROBERT L. MERCER. 1993. Introduction to the special issue on computational linguistics using large corpora. *Computational Linguistics* 19.1–24.

COCHRAN, W. G. 1950. The comparison of percentages in matched samples. *Biometrika* 37.256–266.

COOPER, WILLIAM S. 1969. Is interindexer consistency a hobgoblin? *American Documentation* 20.268–278.

——, FREDRIC C. GEY, & AITOA CHEN. 1994. Probabilistic retrieval in the TIPSTER collections: An application of staged logistic regression. In *Proceedings of TREC-2*, ed. by Donna Harman.

CROFT, W. BRUCE, ROBERT KROVETZ, & H. TURTLE. 1990. Interactive retrieval of complex documents. *Information Processing and Management* 26.593–616.

——, & HOWARD R. TURTLE. 1992. Text retrieval and inference. In *Text-based intelligent systems: Current research and practice in information extraction and retrieval*, ed. by Paul S. Jacobs, 127–156. Lawrence Erlbaum Associates.

CROUCH, C. J. 1990. An approach to the automatic construction of global thesauri. *Information Processing and Management* 26.629–640.

CUTTING, DOUGLAS R., DAVID KARGER, & JAN PEDERSEN. 1993. Constant interaction-time Scatter/Gather browsing of very large document collections. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, 126–135, Pittsburgh, PA.

——, JAN O. PEDERSEN, DAVID KARGER, & JOHN W. TUKEY. 1992. Scatter/Gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th Annual International ACM/SIGIR Conference*, 318–329, Copenhagen, Denmark.

CUTTING, DOUGLASS R., JAN O. PEDERSEN, PER-KRISTIAN HALVORSEN, & MEG WITHGOTT. 1990. Information theater versus information refinery. In *AAAI Spring Symposium on Text-based Intelligent Systems*, ed. by Paul S. Jacobs.

DAGAN, IDO, SHAUL MARCUS, & SHAUL MARKOVITCH. 1993. Contextual word similarity and estimation from sparse data. In *Proceedings of the 31th Annual Meeting of the Association for Computational Linguistics*, 164–171.

DALRYMPLE, MARY, STUART M. SHEIBER, & F PEREIRA. 1991. Ellipsis and higher-order unification. *Linguistics and Philosophy* 14.399–452.

DAVIS, JAMES R. 1994. A server for a distributed digital technical report library. Technical Report Computer Science Report Number 94-1418, Cornell University.

DE TOCQUEVILLE, ALEXIS. 1835. *Democracy in America, Volume I*. London: Saunders and Otley.

DEERWESTER, SCOTT, SUSAN T. DUMAIS, GEORGE W. FURNAS, THOMAS K. LANDAUER, & RICHARD HARSHMAN. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41.391–407.

DEJONG, GERALD F. 1982. An overview of the FRUMP system. In *Strategies for natural language processing*, ed. by Wendy G. Lehnert & Martin H. Ringle, 149–176. Hillsdale: Erlbaum.

FARLEY, LAINE. 1989. Dissecting slow searches. *University of California Division of Library Automation Bulletin (DLA)* 9.

FILLMORE, CHARLES J. 1981. Pragmatics and the description of discourse. In *Radical pragmatics*, ed. by Peter Cole. New York: Academic Press Inc.

FISHER, DAVID, 1994. Topic characterization of full length texts using direct and indirect term evidence. Masters Report, University of California, Berkeley, to appear.

FOWLER, RICHARD H., WENDY A. L. FOWLER, & BRADLEY A. WILSON. 1991. Integrating query, thesaurus, and documents through a common visual representation. In *Proceedings of the 14th Annual International ACM/SIGIR Conference*, 142–151, Chicago.

FOX, EDWARD A., & MATTHEW B. KOLL. 1988. Practical enhanced Boolean retrieval: Experiences with the SMART and SIRE systems. *Information Processing and Management* 24.

FRUCHTERMANN, T., & E. RHEINGOLD. 1990. Graph drawing by force-directed placement. Technical Report UIUCDCS-R-90-1609, Department of Computer Science, University of Illinois, Urbana-Champagne, Ill.

FUHR, NORBERT, & CHRIS BUCKLEY. 1993. Optimizing document indexing and search term weighting based on probabilistic models. In *The first text retrieval conference (TREC-1)*, ed. by Donna Harman, 89–100. NIST Special Publication 500-207.

FULLER, MICHAEL, ERIC MACKIE, RON SACKS-DAVIS, & ROSS WILKINSON. 1993. Coherent answers for a large structured document collection. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, 204–213, Pittsburgh, PA.

FUNG, ROBERT M., STUART L. CRAWFORD, LEE A. APPELBAUM, & RICHARD M. TONG. 1990. An architecture for probabilistic concept-based information retrieval. In *Proceedings of the 13th International ACM/SIGIR Conference*, 455–467.

GALE, WILLIAM A., KENNETH W. CHURCH, & DAVID YAROWSKY. 1992a. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proceedings of the 30th Meeting of the Association for Computational Linguistics*, 249–256.

——, KENNETH W. CHURCH, & DAVID YAROWSKY. 1992b. A method for disambiguating word senses in a large corpus. *Computers and the Humanities* 5-6.415–439.

——, KENNETH W. CHURCH, & DAVID YAROWSKY. 1992c. One sense per discourse. In *Proceedings of the DARPA Speech and Natural Language Workshop*.

GIRILL, T. R. 1991. Information chunking as an interface design issue for full-text databases. In *Interfaces for information retrieval and online systems*, ed. by Martin Dillon, 149–158. New York, NY: Greenwood Press.

GREFENSTETTE, G. 1992. A new knowledge-poor technique for knowledge extraction from large corpora. In *Proceedings of the 15th Annual International ACM/SIGIR Conference*, Copenhagen, Denmark. ACM.

GRIFFITHS, ALAN, H. CLAIRE LUCKHURST, & PETER WILLETT. 1986. Using interdocument similarity information in document retrieval systems. *Journal of the American Society for Information Science* 37.3–11.

GRIMES, J. 1975. *The thread of discourse*. The Hague: Mouton.

GROSZ, BARBARA J. 1986. The representation and use of focus in a system for understanding dialogs. In *Readings in natural language processing*, ed. by Barbara J. Grosz, Karen Sparck Jones, & Bonnie Lynn Webber, 353–362. Morgan Kaufmann.

——, & CANDACE L. SIDNER. 1986. Attention, intention, and the structure of discourse. *Computational Linguistics* 12.172–204.

HAHN, UDO. 1990. Topic parsing: Accounting for text macro structures in full-text analysis. *Information Processing and Management* 26.135–170.

HALLIDAY, M. A. K., & R. HASAN. 1976. *Cohesion in English*. London: Longman.

HARDT, DANIEL. 1992. An algorithm for VP ellipsis. In *Proceedings of the 30th Meeting of the Association for Computational Linguistics*, 9–14.

HARMAN, DONNA. 1993. Overview of the first Text REtrieval Conference. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, 36–48, Pittsburgh, PA.

HAYES, PHILLIP J. 1992. Intelligent high-volume text processing using shallow, domain-specific techniques. In *Text-based intelligent systems: Current research and practice in information extraction and retrieval*, ed. by Paul S. Jacobs, 227–242. Lawrence Erlbaum Associates.

HEARST, MARTI A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics*, 539–545, Nantes, France.

——. 1993. TextTiling: A quantitative approach to discourse segmentation. Technical Report Sequoia 93/24, Computer Science Department, University of California, Berkeley.

——, & CHRISTIAN PLAUNT. 1993. Subtopic structuring for full-length document access. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, 59–68, Pittsburgh, PA.

——, & HINRICH SCHÜTZE. 1993. Customizing a lexicon to better suit a computational task. In *Proceedings of the SIGLEX Workshop on Acquisition of Lexical Knowledge from Text*, 55–69, Columbus, OH.

HENZLER, ROLF G. 1978. Free or controlled vocabularies: Some statistical user-oriented evaluations of biomedical information systems. *International Classification* 5.21–26.

HINDS, JOHN. 1979. Organizational patterns in discourse. In *Syntax and semantics: Discourse and syntax*, ed. by Talmy Givón, volume 12, 135–158. Academic Press.

HIRSCHBERG, JULIA, & DIANE LITMAN. 1993. Empirical studies on the disambiguation of cue phrases. *Computational Linguistics* 19.501–530.

HOBBS, JERRY. 1978. Resolving pronoun references. *Lingua* 44.311–338.

HOVY, ED. 1990. Parsimonious and profligate approaches to the question of discourse structure relations. In *5th ACL Workshop on Natural Language Generation*, Dawson, Pennsylvania.

HWANG, CHUNG HEE, & LENHART K. SCHUBERT. 1992. Tense trees as the 'fine structure' of discourse. In *Proceedings of the 30th Meeting of the Association for Computational Linguistics*, 232–240.

JACOBS, PAUL. 1993. Using statistical methods to improve knowledge-based news categorization. *IEEE Expert* 8.13–23.

——, & LISA RAU. 1990. SCISOR: Extracting information from On-Line News. *Communications of the ACM* 33.88–97.

JURAFSKY, DANIEL, 1992. *An on-line computational model of human sentence interpretation: A theory of the representation and use of linguistic knowledge*. University of California at Berkeley dissertation. (Computer Science Division Report Number 92/676).

JUSTESON, J. S., & S. M. KATZ. 1991. Co-occurrences of antonymous adjectives and their contexts. *Computational Lingustics* 17.1–19.

KAHLE, BREWSTER, & ART MEDLAR. 1991. An information system for corporate users: Wide area information servers. Technical Report TMC199, Thinking Machines Corporation.

KEEN, E. MICHAEL. 1991. The use of term position devices in ranked output experiment. *Journal of Documentation* 47.1–22.

——. 1992. Term position ranking: some new test results. In *Proceedings of the 15th Annual International ACM/SIGIR Conference*, 66–76, Copenhagen, Denmark.

KOLODNER, JANET L. 1983. Maintaining organization in a dynamic long-term memory. *Cognitive Science* 7.243–280.

KORFHAGE, ROBERT R. 1991. To see or not to see – is that the query? In *Proceedings of the 14th Annual International ACM/SIGIR Conference*, 134–141, Chicago.

KOSSLYN, S., S. PINKER, W. SIMCOX, & L. PARKIN. 1983. *Understanding charts and graphs: A project in applied cognitive science*. National Institute of Education. ED 1.310/2:238687.

KOZIMA, HIDEKI. 1993. Text segmentation based on similarity between words. In *Proceedings of the 31th Annual Meeting of the Association for Computational Linguistics*, 286–288, Columbus, OH.

KUNO, SUSUMO. 1972. Functional sentence perspective: A case study from Japanese and English. *Linguistic Inquiry* 3.269–320.

KUPIEC, JULIAN. 1993. MURAX: A robust linguistic approach for question answering using an on-line encyclopedia. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, 181–190, Pittsburgh, PA.

LAKOFF, GEORGE P. 1972. Structural complexity in fairy tales. *The Study of Man* 1.128–150.

LAMBERT, LYNN, & SANDRA CARBERRY. 1991. A tripartite plan-based model of dialogue. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, 47–54.

LANCASTER, F. 1986. *Vocabulary control for information retrieval, second edition*. Arlington, VA: Information Resources.

LARSON, RAY R. 1991. Classification clustering, probabilistic information retrieval, and the online catalog. *The Library Quarterly* 61.133–173.

——. 1992. Experiments in automatic library of congress classification. *Journal of the American Society for Information Science* 43.130–148.

LEWIS, DAVID D. 1992. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th Annual International ACM/SIGIR Conference*, 37–50, Copenhagen.

LIDDY, ELIZABETH. 1991. The discourse level structure of empirical abstracts – an exploratory study. *Information Processing and Management* 27.55–81.

LIDDY, ELIZABETH D., & S. MYAENG. 1993. DR-LINK's linguistic-conceptual approach to document detection. In *The first text retrieval conference (TREC-1)*, ed. by Donna Harman, 113–129. NIST Special Publication 500-207.

——, & WOOJIN PAIK. 1992. Statistically-guided word sense disambiguation. In *Proceedings of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*.

LONGACRE, R. E. 1979. The paragraph as a grammatical unit. In *Syntax and semantics: Discourse and syntax*, ed. by Talmy Givón, volume 12, 115–134. Academic Press.

LUPERFOY, SUSANN. 1992. The representation of multimodal user interface dialogues using discourse pegs. In *Proceedings of the 30th Meeting of the Association for Computational Linguistics*, 22–31.

LYNCH, CLIFFORD. 1992. The next generation of public access information retrieval systems for research libraries – lessons from 10 years of the melvyl system. *Information Technology and Libraries* 11.405–415.

MACKINLAY, JOCK, 1986. *Automatic design of graphical presentations*. Stanford University dissertation. Technical Report Stan-CS-86-1038.

MANBER, UDI, & SUN WU. 1994. GLIMPSE: a tool to search through entire file systems. In *Proceedings of the Winter 1994 USENIX Conference*, 23–31, San Francisco, CA.

MANN, WILLIAM C., & SANDRA A. THOMPSON. 1987. Rhetorical structure theory: A theory of text organization. Technical Report ISI/RS 87-190, ISI.

MANNING, CHRISTOPHER D. 1993. Automatic acquisition of a large subcategorization dictionary from corpora. In *Proceedings of the 31st Annual Meeting of the Association for Computational Lingusitics*, 235–242, Columbus, OH.

MARCHIONINI, GARY, PETER LIEBSCHER, & XIA LIN. 1991. Authoring hyperdocuments: Designing for interaction. In *Interfaces for information retrieval and online systems*, ed. by Martin Dillon, 119–131. New York, NY: Greenwood Press.

MARKEY, KAREN, PAULINE ATHERTON, & CLAUDIA NEWTON. 1982. An analysis of controlled vocabulary and free text search statements in online searches. *Online Review* 4.225–236.

MARKOWITZ, JUDITH, THOMAS AHLSWEDE, & MARTHA EVENS. 1986. Semantically significant patterns in dictionary definitions. *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics* 112–119.

MARTIN, JAMES H. 1990. *A computational model of metaphor interpretation*. Boston: Academic Press.

MASAND, BRIJ, GORDON LINOFF, & DAVID WALTZ. 1992. Classifying news stories using memory based reasoning. In *Proceedings of ACM/SIGIR*, 59–65.

MAULDIN, MICHAEL L., 1989. *Information retrieval by text skimming*. Pittsburg, PA: Carnegie Mellon University dissertation.

——. 1991. Retrieval performance in ferret. In *Proceedings of ACM/SIGIR*, 347–355, Chicago, IL.

MCCUNE, B., R. TONG, J.S. DEAN, & D. SHAPIRO. 1985. Rubric: A system for rule-based information retrieval. *IEEE Transactions on Software Engineering* 11.

MICHARD, A. 1982. Graphical presentation of Boolean expressions in a database query language: design notes and an ergonomic evaluation. *Behaviour and Information Technology* 1.

MILLER, GEORGE A., RICHARD BECKWITH, CHRISTIANE FELLBAUM, DEREK GROSS, & KATHERINE J. MILLER. 1990. Introduction to WordNet: An on-line lexical database. *Journal of Lexicography* 3.235–244.

MINSKY, MARVIN. 1975. A framework for representing knowledge. In *The psychology of computer vision*, ed. by Patrick Winston. McGraw-Hill.

MITTENDORF, ELKE, & PETER SCHÄUBLE. 1994. Passage retrieval based on hidden markov models. In *Proceedings of the 17th Annual International ACM/SIGIR Conference*, Dublin, Ireland. To appear.

MOFFAT, ALISTAIR, RON SACKS-DAVIS, ROSS WILKINSON, & JUSTIN ZOBEL. 1994. Retrieval of partial documents. In *Proceedings of TREC-2*, ed. by Donna Harman. To appear.

MOONEY, DAVID J., M. SANDRA CARBERRY, & KATHLEEN F. MCCOY. 1990. The generation of high-level structure for extended explanations. In *Proceedings of the Thirteenth International Conference on Computational Linguistics*, volume 2, 276–281, Helsinki.

MOORE, JOHANNA D., & MARTHA E. POLLACK. 1992. A problem for RST: The need for multi-level discourse analysis. *Computational Linguistics* 18.

MORRIS, JANE. 1988. Lexical cohesion, the thesaurus, and the structure of text. Technical Report CSRI-219, Computer Systems Research Institute, University of Toronto.

——, & GRAEME HIRST. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics* 17.21–48.

NOREAULT, TERRY, MICHAEL MCGILL, & MATTHEW B. KOLL. 1981. A performance evaluation of similarity measures, document term weighting schemes and representations in a Boolean environment. In *Information retrieval research*, ed. by R. N. Oddy, S. E. Robertson, C. J. van Rijsbergen, & P. W. Williams, 57–76. London: Butterworths.

NORVIG, PETER, 1987. *A unified theory of inference for text understanding*. University of California, Berkeley dissertation. (Computer Science Division Report Number 87/339).

O'CONNOR, J. 1980. Answer passage retrieval by text searching. *Journal of the ASIS* 32.227–239.

OUSTERHOUT, JOHN. 1991. An X11 toolkit based on the Tcl language. In *Proceedings of the Winter 1991 USENIX Conference*, 105–115, Dallas, TX.

PAICE, CHRIS D. 1990. Constructing literature abstracts by computer: Techniques and prospects. *Information Processing and Management* 26.171–186.

PASSONNEAU, REBECCA J., & DIANE J. LITMAN. 1993. Intention-based segmentation: Human reliability and correlation with linguistic cues. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, 148–155.

PEAT, HELEN J., & PETER WILLETT. 1991. The limitations of term co-occurence data for query expansion in document retrieval systems. *JASIS* 42.378–383.

PHILLIPS, MARTIN. 1985. *Aspects of text structure: An investigation of the lexical organisation of text*. Amsterdam: North-Holland.

POLLOCK, J. J., & A. ZAMORA. 1975. Automatic abstracting research at Chemical Abstracts Service. *Journal of Chemical Information and Computer Sciences* 15.226–233.

PUSTEJOVSKY, JAMES. 1987. On the acquisition of lexical entries: The perceptual origin of thematic relations. *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics* .

RASKIN, VICTOR, & IRWIN WEISER. 1987. *Language and writing: Applications of linguistics to rhetoric and composition*. Norwood, New Jersey: ABLEX Publishing Corporation.

RESNIK, PHILIP. 1992. WordNet and Distributional Analysis: A Class-based Approach to Lexical Discovery. In *Statistically-based natural language programming techniques: Papers from the 1992 workshop*, ed. by Carl Weir. Menlo Park, CA: AAAI Press,Technical Report W-92-01.

——, 1993. *Selection and information: A class-based approach to lexical relationships*. University of Pennsylvania dissertation. (Institute for Research in Cognitive Science report IRCS-93-42).

RILOFF, ELLEN, & WENDY LEHNERT. 1992. Classifying texts using relevancy signatures. In *Proceedings of the Tenth National Conference on Artificial Intelligence*. AAAI Press/The MIT Press.

RO, JUNG SOON. 1988a. An evaluation of the applicability of ranking algorithms to improve the effectiveness of full-text retrieval. I. On the effectiveness of full-text retrieval. *Journal of the American Society for Information Science* 39.73–78.

——. 1988b. 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. *Journal of the American Society for Information Science* 39.147–160.

ROBERTSON, GEORGE C., STUART K. CARD, & JOCK D. MACKINLAY. 1993. Information visualization using 3D interactive animation. *Communications of the ACM* 36.56–71.

ROLLING, L. 1981. Indexing consistency, quality, and efficiency. *Information Processing and Management* 17.69–76.

ROSE, DANIEL E., & RICHARD K. BELEW. 1991. Toward a direct-manipulation interface for conceptual information retrieval systems. In *Interfaces for information retrieval and online systems*, ed. by Martin Dillon, 39–54. New York, NY: Greenwood Press.

ROTONDO, JOHN A. 1984. Clustering analysis of subjective partitions of text. *Discourse Processes* 7.69–88.

RUGE, GERDA. 1991. Experiments on linguistically based term associations. In *Proceedings of the RIAO*, 528–545.

RUMELHART, DAVID. 1975. Notes on a schema for stories. In *Representation and understanding*, ed. by Daniel G. Bobrow & Allan Collins, 211–236. New York: Academic Press.

RUS, DANIELA, & DEVIKA SUBRAMANIAN. 1993. Multi-media RISSC informatics: Retrieving information with simple structural components. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*.

SALTON, GERARD (ed.) 1971. *The Smart retrieval system – experiments in automatic document processing*. Englewood Cliffs, NJ: Prentice Hall.

——. 1972. Experiments in automatic thesaurus construction for information retrieval. In *Information Processing 71*, 115–123. North Holland Publishing Co.

——. 1988. *Automatic text processing : the transformation, analysis, and retrieval of information by computer*. Reading, MA: Addison-Wesley.

——, JAMES ALLAN, & CHRIS BUCKLEY. 1993. Approaches to passage retrieval in full text information systems. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, 49–58, Pittsburgh, PA.

——, JAMES ALLAN, & CHRIS BUCKLEY. 1994. Automatic structuring and retrieval of large text files. *Communications of the ACM* 37.97–108.

——, & CHRIS BUCKLEY. 1990. Improving retrieval performance by relevance feedback. *JASIS* 41.288–297.

——, & CHRIS BUCKLEY. 1991. Automatic text structuring and retrieval: Experiments in automatic encyclopedia searching. In *Proceedings of the 14th Annual International ACM/SIGIR Conference*, 21–31.

——, & CHRIS BUCKLEY. 1992. Automatic text structuring experiments. In *Text-based intelligent systems: Current research and practice in information extraction and retrieval*, ed. by Paul S. Jacobs, 199–209. Lawrence Erlbaum Associates.

SCHANK, ROGER, & R. ABELSON. 1977. *Scripts, plans, goals, and understanding*. Hillsdale, NJ: Erlbaum.

SCHIFFRIN, DEBORAH. 1987. *Discourse markers*. Cambridge: Cambridge University Press.

SCHÜTZE, HINRICH. 1993a. Part-of-speech induction from scratch. In *Proceedings of ACL 31*, Ohio State University.

——. 1993b. Word space. In *Advances in neural information processing systems 5*, ed. by Stephen J. Hanson, Jack D. Cowan, & C. Lee Giles. San Mateo CA: Morgan Kaufmann.

SENAY, HIKMET, & EVE IGNATIUS. 1990. Rules and principles of scientific data visualization. Technical Report GWU-IIST-90-13, Institute for Information Science and Technology, The George Washington University.

SHNEIDERMAN, BEN. 1987. *Designing the user interface: strategies for effective human-computer interaction*. Reading, MA: Addison-Wesley.

SIBUN, PENELOPE. 1992. Generating text without trees. *Computational Intelligence: Special Issue on Natural Language Generation* 8.102–122.

SIDNER, CANDACE L. 1983. Focusing in the comprehension of definite anaphora. In *Computational models of discourse*, ed. by Michael Brady & Robert C. Berwick, 267–330. Cambridge, MA: MIT Press.

SKOROCHOD'KO, E.F. 1972. Adaptive method of automatic abstracting and indexing. In *Information Processing 71: Proceedings of the IFIP Congress 71*, ed. by C.V. Freiman, 1179–1182. North-Holland Publishing Company.

SMADJA, FRANK A., & KATHLEEN R. MCKEOWN. 1990. Automatically extracting and representing collocations for language generation. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, 252–259.

SPARCK-JONES, KAREN. 1971. *Automatic keyword classification for information retrieval*. London: Butterworth & Co.

——. 1986. *Synonymy and semantic classification*. Edinburgh: Edinburgh University Press.

SPOERRI, ANSELM. 1993. InfoCrystal: A visual tool for information retrieval & management. In *Proceedings of Information Knowledge and Management '93*, Washington, D.C.

STANFILL, CRAIG, & DAVID L. WALTZ. 1992. Statistical methods, artificial intelligence, and information retrieval. In *Text-based intelligent systems: Current research and practice in information extraction and retrieval*, ed. by Paul S. Jacobs, 215–226. Lawrence Erlbaum Associates.

STARK, HEATHER. 1988. What do paragraph markers do? *Discourse Processes* 11.275–304.

STODDARD, SALLY. 1991. *Text and texture: Patterns of cohesion*, volume XL of *Advances in Discourse Processes*. Norwood, NJ: Ablex Publishing Corporation.

STONEBRAKER, MICHAEL, & G. KEMNITZ. 1991. The POSTGRES next-generation database management system. *Communications of the ACM* 34.78–92.

SUNDHEIM, BETH. 1990. Second message understanding conference(MUC-II). Technical Report 1328, Naval Ocean Systems Center, San Diego, CA.

SVENONIUS, ELAINE. 1986. Unanswered questions in the design of controlled vocabularies. *Journal of the American Society for Information Science* 37.331–340.

TANNEN, DEBORAH. 1984. *Conversational style: Analyzing talk among friends*. Norwood, NJ: Ablex.

——. 1989. *Talking voices: Repetition, dialogue, and imagery in conversational discourse*. Studies in Interactional Sociolinguistics 6. Cambridge University Press.

TENOPIR, CAROL, & JUNG SOON RO. 1990. *Full text databases*. New Directions in Information Management. Greenwood Press.

THOMPSON, R. H., & B. W. CROFT. 1989. Support for browsing in an intelligent text retrieval system. *International Journal of Man [sic] -Machine Studies* 30.639–668.

TOMBAUGH, J., A. LICKORISH, & P. WRIGHT. 1987. Multi-window displays for readers of lengthy texts. *International Journal of Man [sic] -Machine Studies* 26.597–615.

TUFTE, EDWARD. 1983. *The visual display of quantitative information*. Chelshire, CT: Graphics Press.

VAN DIJK, TEUN A. 1980. *Macrostructures*. Hillsdale, N.J.: Lawrence Erlbaum Associates.

——. 1981. *Studies in the pragmatics of discourse*. The Hague: Mouton Publishers.

VAN RIJSBERGEN, C. J. 1979. *Information retrieval*. London: Butterworths.

VOORHEES, ELLEN M. 1985. The cluster hypothesis revisited. In *Proceedings of ACM/SIGIR*, 188–196.

WALKER, MARILYN. 1991. Redundancy in collaborative dialogue. In *AAAI Fall Symposium on Discourse Structure in Natural Language Understanding and Generation*, ed. by Julia Hirschberg, Diane Litman, Kathy McCoy, & Candy Sidner, Pacific Grove, CA.

WANG, MICHELLE Q., & JULIA HIRSCHBERG. 1992. Automatic classification of intonational phrase boundaries. *Computer Speech and Language* 6.175–196.

WEBBER, BONNIE LYNN. 1987. The interpretation of tense in discourse. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, 147–154.

WILENSKY, ROBERT. 1981. Meta-planning: representing and using knowledge about planning in problem solving and natural language understanding. *Cognitive Science* 5.197–235.

——. 1983a. *Planning and understanding*. Reading, MA: Addison-Wesley.

——. 1983b. Story grammars vs. story points. *The Behavior and Brain Sciences* 6.

——, YIGAL ARENS, & DAVID N. CHIN. 1984. Talking to UNIX in English: An overview of UC. *Communications of the ACM* 27.

WILKS, YORICK. 1975. An intelligent analyzer and understander of English. *Communications of the ACM* 18.264–274.

WILKS, YORICK A., DAN C. FASS, CHENG MING GUO, JAMES E. MCDONALD, TONY PLATE, & BRIAN M. SLATOR. 1990. Providing machine tractable dictionary tools. *Journal of Computers and Translation* 2.

WINOGRAD, TERRY. 1972. *Understanding natural language*. New York, NY: Academic Press.

WU, SUN, & UDI MANBER. 1992. Agrep – a fast approximate pattern-matching tool. In *Proceedings of the Winter 1992 USENIX Conference*, 153–162, San Francisco, CA.

YAROWSKY, DAVID. 1992. Word sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics*, 454–460, Nantes, France.

YOUMANS, GILBERT. 1991. A new tool for discourse analysis: The vocabulary-management profile. *Language* 67.763–789.