Experiments in Automatic Phrase Indexing For Document Retrieval: A Comparison of Syntactic and Non-Syntactic Methods

Joel L. Fagan
Ph.D. Thesis

87-868
September 1987

Department of Computer Science
Cornell University
Ithaca, New York 14853-7501

EXPERIMENTS IN AUTOMATIC PHRASE INDEXING FOR DOCUMENT RETRIEVAL: A COMPARISON OF SYNTACTIC AND NON-SYNTACTIC METHODS

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Joel L. Fagan

January 1988

EXPERIMENTS IN AUTOMATIC PHRASE INDEXING FOR DOCUMENT RETRIEVAL: A COMPARISON OF SYNTACTIC AND NON-SYNTACTIC METHODS

Joel L. Fagan, Ph.D.
Cornell University 1988

In order for an automatic information retrieval system to effectively retrieve documents related to a given subject area, the content of each document in the system's database must be represented accurately. This study examines the hypothesis that better representations of document content can be constructed if the content analysis method takes into consideration the syntactic structure of document and query texts. Two methods of automatically generating phrases for use as content indicators have been implemented and tested experimentally. The non-syntactic (or statistical) method is based on simple text characteristics such as word frequency and the proximity of words in text. The syntactic method uses augmented phrase structure rules (production rules) to selectively extract phrases from parse trees generated by an automatic syntactic analyzer.

Experimental results show that the effect of non-syntactic phrase indexing is inconsistent. For the five collections tested, increases in average precision ranged from 22.7% to 2.2% over simple, single term indexing. The syntactic phrase indexing method was tested on two collections. Precision

figures averaged over all test queries indicate that non-syntactic phrase indexing performs significantly better than syntactic phrase indexing for one collection, but that the difference is insignificant for the other collection. More detailed analysis of individual queries, however, indicates that the performance of both methods is highly variable, and that there is evidence that syntax-based indexing has certain benefits not available with the non-syntactic approach.

Possible improvements of both methods of phrase indexing are considered. It is concluded that the prospects for improving the syntax-based approach to document indexing are better than for the non-syntactic approach.

The PLNLP system was used for syntactic analysis of document and query texts, and for implementing the syntax-based phrase construction rules. The SMART information retrieval system was used for retrieval experimentation.

This thesis is available as a technical report from the Department of Computer Science, Cornell University.

BIOGRAPHICAL SKETCH

Joel L. Fagan was born on February 6, 1952 in Orofino, Idaho. He graduated from Marshfield Senior High School, Coos Bay, Oregon in 1970. In 1975 he received a Bachelor of Arts degree in Anthropology, with Distinction and Honors, from Northwestern University in Evanston, Illinois; he became a member of Phi Beta Kappa the same year. In 1979 he completed the degree of Master of Arts in Linguistics at the University of Hawaii at Manoa in Honolulu. He enrolled in the Ph.D. program in Linguistics at Cornell University in 1980.

To emacs, dudley, and Truck.

ACKNOWLEDGMENTS

The context in which things happen often has a strong effect on what happens and how. But the context often goes unnoticed because it appears to provide only background. Cornell University has a broad spectrum of high quality resources that provide a productive context for many scholarly and scientific endeavors. Having access to these resources is a pleasure that I appreciate greatly.

I am fortunate to have been associated with two departments while at Cornell. My official connection is with the Department of Modern Languages and Linguistics. From that department, Linda R. Waugh, Chair of my Special Committee, has been a willing and competent adviser on all matters throughout the varied course of my graduate program. Because of her excellent teaching, and her careful treatment of various linguistic phenomena that are left untouched in many linguistics programs, my understanding of certain aspects of linguistic structure has been broadened considerably. John U. Wolff is a member of my Special Committee representing a minor in Southeast Asian Linguistics. I appreciate and have benefited from his efforts to provide a strong program in Indonesian at Cornell, as well as opportunities for language study abroad. Other members of the linguistics community at Cornell who have been helpful to me in various ways include Richard L. Leed, Susan Hertz, and Joseph E. Grimes.

The Department of Computer Science is where I have done nearly all of the work related to this thesis. Gerard Salton is my Thesis Adviser and a member of my Special Committee representing a minor in Computer Science. He has made it possible for me to do a great number of things that almost certainly could not have been done without his help. His extensive knowledge of and experience in the field of information retrieval makes his advice very valuable. It would be difficult to overstate the extent of the benefit I have derived from being associated with Professor Salton and his research group. For this opportunity, I am very grateful.

I have also benefited from contact with several people who have been regular participants in the Information Retrieval Seminar. These include: Chris Buckley, Ellen Voorhees, Maria Smith, José Araya, C. D. Paice (Visiting Professor from University of Lancaster, U.K.), Carolyn Crouch, and Donald Crouch (both Visiting Professors, now at Tulane University). Ellen Voorhees and Chris Buckley deserve further thanks for introducing me to the department's computing environment, and getting me started in the business of retrieval experimentation. I have learned a lot about information retrieval and a variety of other computing-related matters from both of them.

I owe an additional measure of appreciation to Chris Buckley, since it was his suggestion that set me to work on the problem of phrase indexing. Also, the current implementation of the SMART experimental information retrieval system would not exist in its present form without the benefits of

Buckley's many talents and persistence. Without this well-designed system, the experimental work done for this thesis would have been much more difficult, and much less extensive.

The Department of Computer Science as a whole has been exceedingly generous in allowing me to have access to its excellent facilities. It has been a great pleasure to have essentially unlimited use of the department's computing equipment. In addition, I have been provided with personal work space: a desk and a chair in a comfortable office with a telephone that has access to long distance lines. Even a personal mailbox has been provided. These luxuries are entirely unknown to graduate students in less fortunate departments on campus.

The practicalities of day-to-day life in a large organization are much easier to deal with if you know exactly where to go or who to call to solve whatever problem has just arisen. Geraldine L. Pinkham, Accounts Coordinator for the Department of Computer Science, is a veritable fountain of knowledge regarding crucial matters of this kind. Her abilities and willingness to help are typical of the administrative and office staff in the department.

There is a lot of computing equipment in the Department of Computer Science, and the CER staff has the responsibility of keeping it all running. I am indebted to the entire staff for doing an admirable job of keeping on top of this never-ending task. Because he has been the target of most of my ques-

tions and complaints, Larry Parmelee deserves special thanks. I have learned a number of useful things from him.

Though Cornell has a lot to offer, it doesn't have everything. This has given me an excuse to make some contacts that have proven to be as important and beneficial as those I have made here at Cornell. During the summer of 1985, I had the good fortune of working with George E. Heidorn and Karen Jensen at IBM's T. J. Watson Research Center. The syntactically oriented work that appears in this thesis is a direct outgrowth of work that I started there. Their interest and encouragement, and their willingness to teach me how to use the PLNLP system and to make the system available for installation at Cornell are appreciated greatly. Stephen D. Richardson, also a member of Heidorn's research group, has been very helpful in matters related to using the dictionary associated with the PLNLP system.

I will close my acknowledgements of individuals by reaching a bit further back. During the years I spent acquiring a reputation as an itinerant student, I encountered a few people who have influenced me significantly. Stanley Starosta's views on syntax have left a lasting impression. From John Terrell, I learned a lot about scholarship and science. Mark Papworth first convinced me that it might be worthwhile to look around the next bend in the road. Bruce Martin and John Johnson gave me some essential mathematical skills that continue to serve me well.

In the foregoing, I have restricted my expressions of appreciation to my Little Britain sentiments, leaving my Walworth sentiments to be conveyed more directly. This choice has nothing to do with the relative importance of the contributions of the individuals involved in these two realms of experience. In fact, the contributions of the family members and friends that figure in the Walworth sentiments surely out-shine the others by several magnitudes of brightness.

During the course of my graduate program, financial support has come from several sources; these include: (1) the National Science Foundation, grants IST 83-16166 and IST 85-44189, to Cornell University, Gerard Salton, principal investigator, (2) two grants from OCLC, Inc. to Cornell University, Gerard Salton, principal investigator, (3) Teaching Assistantships from the Department of Computer Science and the Department of Modern Languages and Linguistics, (4) a Teaching Associateship (Indonesian) from the Department of Modern Languages and Linguistics, (5) National Resource Fellowships for Indonesian, administered by the Southeast Asia Program, and (6) two travel grants from the Graduate School, with supplementary funding from the Field of Linguistics and the Southeast Asia Program. For their efforts in making this support available to me, I am grateful to Leonard H. Babby, George N. Clements, Sheila J. Haddad, Michael J. McGill, Stanley J. O'Connor, Gerard Salton, Charles VanLoan, Linda R. Waugh, and John U. Wolff.

Other forms of support are also gratefully acknowledged. Some of the computations supporting this research were performed at the Cornell National Supercomputer Facility, which is supported in part by the National Science Foundation, New York State, and IBM Corporation. IBM Corporation also provided the PLNLP natural language processing system, which is the software used in this research for all processing related to syntactic analysis.

TABLE OF CONTENTS

Three: **Syntactic Phrase Indexing** **74**

Four: **Comparison of Phrase Indexing Experiments** 183

Five: **Conclusion** 237

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The purpose of a document retrieval system is to respond to a request for information about a particular topic by returning to the user a list of references to documents that are related to that topic. An important step in this process is content analysis. In fully automatic systems, content analysis involves scanning the text of a document and extracting items that are expected to be good indicators of the document's content. These content indicators are then used to construct a reduced representation of the document.

To a person interested in knowing what a particular document is about, it is more informative to know, for example, that the phrase *computer science* is present in the document than it is to know that the word *computer* and the word *science* both occur in the document. It is easy to see this by observing that the pair of disassociated terms, *computer* and *science* characterize titles (1.1) and (1.2) equally well, while the phrase *computer science* is applicable only to (1.2).

(1.1) New *Computer* Technology and its Impact on Materials *Science*

(1.2) The Undergraduate Curriculum in *Computer Science*

Similarly, in a document retrieval system, the representation of a document containing the phrase *computer science* would be more accurate if it included the phrase rather than the corresponding pair of disassociated words in its set of content indicators. A query containing such a phrase could then match on documents like (1.2), but avoid matching on documents like (1.1).

This simple example illustrates an obvious shortcoming of the document representation models used in most automatic systems. In such systems, the content of each document is represented by an unstructured collection of simple descriptors (single words or word stems). The document representations typically do not include any indication of syntactic or semantic relationships among words in text. In addition, statistical independence of terms is generally assumed. Simplified representations of this kind reduce the accuracy of the representations of document content. Inaccuracies in content representation can be expected to inhibit the effectiveness of the retrieval system.

The general problem addressed by this study is that of improving the quality of automatic methods of text analysis and representation of document content. The point of view taken in examining this problem is that

(a) the quality of document content analysis and representation should have a substantial influence on the overall effectiveness of a document retrieval system, and

(b) better representations of document content can be constructed if the content analysis method takes into consideration information about the structure of document and query texts.

There are many aspects of text structure that could be useful for the task of content analysis and representation. These include, for example, identification of case relations or other functional relationships (Sparck Jones and Tait 1984a, 1984b; Lewis and Croft 1987; Di Benigno, Cross, and deBessonet 1986; Reeker, Zamora, and Blower 1983), recognition of anaphoric elements (Liddy et al. 1987), and determining other discourse relations among text elements (Strong 1973, 1974; Liddy 1987). Accurate automatic analysis of these more complex aspects of text structure is, however, beyond the capabilities of current natural language processing technology, at least for the large volumes of unrestricted text that must be dealt with by general-purpose document retrieval systems (DeJong 1983). Similarly, methods that are intended for use only in narrowly restricted domains, and which depend, for example, on the use of sublanguage grammars, detailed representations of domain knowledge (often constructed by hand), or specially structured document collections are also not applicable.[1]

An aspect of text structure that should be useful for purposes of content analysis, and that may be simple enough to be dealt with automatically, is identification of relationships of modification between words. Relationships of modification are relationships such as those expressed by phrases. In general, the objective of using phrases as content indicators is to take advantage of the fact that phrases identify concepts that are more specific than the concepts identified by their components in isolation. This was illustrated by the examples in (1.1) and (1.2), above. Use of phrases as content indicators is expected to improve the effectiveness of a document retrieval system by enhancing the

[1] Examples of such systems include: Cooper (1984), Cowie (1983), Di Benigno, Cross, and deBessonet (1986), Hahn and Reimer (1985), Lebowitz (1983), Sager (1975, 1981), Schank, Kolodner and DeJong (1981), Tuttle et al. (1983), Vickery, Brooks and Robinson (1987), and Walker and Hobbs (1981).

precision of searches.

With the objective of taking advantage of information of this kind, a number of methods have been proposed for identifying important relationships among words in text and incorporating information about these relationships into document retrieval models. Primary efforts in this area include statistical association methods,[2] probabilistic term dependency models,[3] and recognition of syntactic relationships as a basis for identifying phrases for use as content indicators.[4] In spite of the substantial effort devoted to this general problem, however, there is still no well-established consensus regarding the way in which information about term relationships should be obtained and incorporated into document retrieval systems, or the extent to which this kind of information can be expected to yield consistently positive results in an operational setting. In particular, the potential value of automatic syntactic analysis as a component of a document content analysis system appears to be an open question.

The idea of using linguistic methods for purposes of content analysis surfaced quite early in the development of automated indexing systems and formal models of natural language grammars. In 1958, Zellig Harris suggested

---

[2] See: Stiles (1961), Doyle (1961, 1962), Giuliano and Jones (1963), Salton (1968), and Lesk (1969).

[3] See: van Rijsbergen (1977), Harper and van Rijsbergen (1978), Yu et al. (1983), Salton, Buckley and Yu (1983).

[4] See: Baxendale (1958, 1961), Salton (1966), Earl (1970, 1972), Hillman and Kasarda (1969), Hillman (1973), Klingbiel (1973a, 1973b), Dillon and Gray (1983), Metzler et al. (1984), Aladesulu (1985), Smeaton (1986).

the application of syntactic analysis to content analysis in information retrieval (Harris 1959). Some of these ideas were quickly incorporated into experiments in automatic indexing and abstracting (Climenson, Hardwick and Jacobson 1961).

Some experimentation with linguistic methods, and repeated speculation about the proper use and potential benefits of the application of linguistic methods in content analysis have continued into the current decade. The consensus of those who have considered the issue is that the bond between the fields of linguistics and information science should be a close and mutually beneficial one. This is the point of view expressed by Christine Montgomery (1972:195):

> In theory, the relationship between linguistics and information science is clear and indisputable: information science is concerned with all aspects of the communication of information, language is the primary medium for the communication of information, and linguistics is the study of language as a system for communicating information.

Jean-Claude Gardin (1973) takes a similar position.

Both of these writers, as well as others, point out that syntactic analysis, in the absence of correspondingly sophisticated semantic information, may not be sufficient to provide significant improvement in content analysis (Walker 1981:351-352; Sparck Jones and Kay 1973:4). But in spite of this, the point of view that further experimentation with syntactic analysis in indexing is justified is well represented in the literature. The predominant conclusions are that:[5]

(1) Very little research has been done to determine how to use syntactic information in document analysis, what kinds of syntactic information can be usefully incorporated into document representations, or how retrieval effectiveness is affected by the use of this information.

(2) Of the retrieval experimentation that has been done, the scale has been so small that strong conclusions with respect to the value of syntactic analysis cannot be drawn.

(3) The question of the value of syntactic analysis in content analysis and retrieval remains unresolved, and therefore, additional research in this area should be of interest.

That these conclusions are still applicable at the present time is evidenced by a recent collection of essays that reviews virtually all major experimental information retrieval work since the late 1950s (Sparck Jones 1981). Only one experimental study discussed in that volume involved syntactic methods (Salton 1981).

The objective of this study has been to evaluate one of the more successful existing methods of automatic phrase indexing and then to develop and test a method for constructing phrase descriptors based on automatic syntactic analysis of the text of documents and queries.

1.1. Overview

The remainder of this chapter treats some relevant preliminary matters. Section 1.2 describes the vector space model of information retrieval, which is

---

[5] See: Montgomery (1972:196, 199, 203), Sparck Jones and Kay (1973:105, 106, 111, 112, 118-119), Sparck Jones and Kay (1977:189), Sparck Jones (1974:399, 405, 427, 428), Salton and McGill (1983:287), Sparck Jones and Tait (1984a:50), Croft (1986b:205).

the model used for the experimental work presented in this thesis. Basic characteristics of the experimental document and query collections also appear in that section. Section 1.3 presents in more detail the motivation for phrase indexing. This includes a discussion of term specificity and term relationships (or term associations), and their roles in the problem of phrase indexing. A brief overview of typical ways in which term relationships are dealt with in retrieval systems is also presented. Automatic methods for identifying phrases in the natural language text of documents and queries are discussed in section 1.4.

Chapter 2 examines the effectiveness of the discrimination value model of phrase indexing. This non-syntactic approach to phrase indexing was chosen since the available experimental evidence indicates that it is one of the most effective automatic phrase indexing methods proposed so far (Salton, Yang, and Yu 1975). The objective of this evaluation is to determine the level of effectiveness achievable using non-syntactic phrase indexing. This will make it possible to evaluate the relative effectiveness of syntactic and non-syntactic approaches. Several problems related to phrase indexing are also discussed in this chapter, and possible solutions are proposed that depend on the incorporation of syntactic information into the phrase construction process.

Chapter 3 proposes a syntax-based approach to phrase indexing, and evaluates its effectiveness based on the results of retrieval experiments. Discussion of the phrase indexing method includes a brief overview of the natural language processing system and computational grammar that it is based on. Various strategies for generating phrases from the syntactic structures provided by the syntactic analyzer are introduced and illustrated with examples, and shortcomings of the method are examined. The chapter concludes with a discussion of the results of retrieval experiments.

Chapter 4 compares the syntactic and non-syntactic phrase indexing methods presented in chapters 2 and 3 with regard to their influences on retrieval effectiveness. In addition, both the syntactic and non-syntactic phrase indexing methods examined in this study are compared to previous experimental work on phrase indexing in document retrieval.

Chapter 5 summarizes the experimental results of the preceding chapters, and assesses the general usefulness of both the syntactic and non-syntactic approaches to phrase indexing. Possible refinements of both phrase construction methods are discussed. Finally, a few suggestions are made indicating how the syntax-based approach to phrase construction could be extended to encompass the general task of document content analysis (rather than just phrase construction) using linguistically oriented methods.

1.2. The Experimental Retrieval Environment

1.2.1. The Document Representation and Retrieval Model

The vector space model is the document representation and retrieval model used in this study. In this model, the document collection is represented by document vectors D_i, each identified by one or more descriptors, T_j. Each document is thus represented by a t-dimensional vector. A query vector is represented in the same way:

$$D_i = (d_{i1}, d_{i2}, \cdots d_{it}) \tag{1.3}$$

$$Q = (q_1, q_2, \cdots q_t) \tag{1.4}$$

The elements of the vectors (d_{ij}, q_j) represent the weight, or importance, of the jth descriptor in the query or ith document (Salton, Wong, and Yang 1975; Salton 1975b). A refinement of this simple vector space model has been proposed by Fox (1983a, 1983b). His model employs "extended vectors," in which a complete document or query vector may contain multiple subvectors, each representing a different kind of information. The use of extended vectors for phrase indexing is discussed fully in chapter 2.

In order to illustrate this representation scheme, the two documents and one query in Figure 1.1 are shown in their corresponding vector forms in Figure 1.2.

Document 1

Information Flow in Research and Development Libraries

Document 2

Acquisition Planning in Research and Development Libraries

Query

acquisition in research and development libraries

FIGURE 1.1. Text of sample documents and query.

Descriptor	Descriptor Number	Documents 1	2	...	n	Query
information	1	1	0	...	.	0
flow	2	1	0		.	0
research	3	1	1		.	1
development	4	1	1			1
libraries	5	1	1			1
acquisition	6	0	1			1
planning	7	0	1			0
.	.	.	.			.
.	.	.	.			.
.	t	.	.			.

FIGURE 1.2. Vector representation of sample documents and query.

Given this representation, a function can be defined that reflects the degree of similarity between a pair of vectors. A commonly used similarity function is the cosine correlation (1.5), which is an inverse function of the angle between a pair of vectors.

$$cosine(Q,D_i) = \frac{\sum_{j=1}^{t} (q_j \cdot d_{ij})}{\sqrt{\sum_{j=1}^{t} q_j^2 \cdot \sum_{j=1}^{t} d_{ij}^2}} \tag{1.5}$$

Many other similarity functions have also been proposed and tested (Noreault, McGill, and Koll 1981).

The retrieval process consists of three steps: (1) calculating the similarity between a query and each document in the collection,[6] (2) ranking the documents in decreasing order by similarity value, and (3) returning to the user a specified number of the highest ranking documents.

The vector space model is useful because it provides certain capabilities that are not available with the Boolean model of information retrieval, which is the predominant model used in commercial document retrieval services (Salton 1975a:121-123). In addition, experimental work has shown that the fully automatic indexing and retrieval procedures available with the SMART system, which is based on the vector space model, can yield better retrieval performance than the MEDLARS system, which is based on the Boolean model and uses manual indexing with a controlled indexing vocabulary (Salton 1972a). In summary, the major advantages of the vector space model are:

(1) Query formulation is simplified, since queries need not be stated as expressions in Boolean algebra. A query appropriate for the vector space model can easily be constructed automatically from a natural language statement of a user's information need.

---

[6] A number of methods have been proposed to overcome the inefficiency of sequentially processing the entire document collection. See, for example, Buckley and Lewit (1985), Voorhees (1985), Smeaton and van Rijsbergen (1981), and Salton (1971).

(2) Document and query terms can easily be weighted to reflect their relative importance as indicators of document content.

(3) The basic principle of calculating a similarity coefficient between a query and document makes it possible to:

(a) rank the retrieved documents in decreasing order of similarity with the query, so that the potentially most relevant documents can be presented to the user first,

(b) retrieve documents that only approximately match the query, that is, that contain some, but not all, of the terms in the query, and

(c) easily control the number of documents returned to the user.

1.2.2. Experimental Document and Query Collections

The phrase indexing and retrieval experiments to be described in chapters 2 and 3 use five document and query collections: CACM, CISI, CRAN, MED, and INSPEC. The CACM collection contains all articles published in the *Communications of the Association for Computing Machinery* in the years 1958-1979. This is a total of 3,204 documents. The CISI collection contains 1,460 documents dealing primarily with information and library science published between 1969 and 1977. The 1,398 documents in the CRAN collection have to do with aerodynamics and aeronautical engineering. This collection is based on one used for the Aslib-Cranfield Project (Cleverdon and Mills 1963; Cleverdon, Mills, and Keen 1966). MED is a selection of 1,033 documents on medicine taken from the National Library of Medicine. The largest collection, INSPEC, contains 12,684 documents on electronics, electrical engineering, and computer science. Basic statistics for these document and associated query collections appear in Table 1.1.

Document Collections		CACM	CISI	CRAN	MED	INSPEC
Number of Documents		3204	1460	1398	1033	12684
Number of Stem Types		4522	5019	3763	6927	14255
Mean Stems per Document		20.22	45.20	53.13	51.60	30.01
Mean Term Frequency	Maximum	2.94	5.29	5.81	5.88	3.86
	Minimum	1.00	1.00	1.00	1.00	1.00
	Mean	1.23	1.39	1.54	1.51	1.36
Mean Document Frequency	Maximum	904.7	573.0	775.7	310.7	3724.5
	Minimum	14.0	2.9	3.3	1.3	14.9
	Mean	236.5	123.2	173.0	59.9	722.1
Query Collections		**CACM**	**CISI**	**CRAN**	**MED**	**INSPEC**
Number of Queries		52	76	225	30	77
Number of Stem Types		324	657	585	241	576
Mean Stems per Query		10.67	22.59	9.17	10.10	15.81
Mean Term Frequency	Maximum	1.98	3.38	1.28	1.53	2.64
	Minimum	1.00	1.00	1.00	1.00	1.00
	Mean	1.14	1.25	1.03	1.08	1.21
Mean Document Frequency	Maximum	754.1	581.2	532.5	190.1	3371.5
	Minimum	17.6	21.9	31.3	8.5	45.0
	Mean	205.6	186.2	197.2	59.1	752.3

TABLE 1.1. Statistics for document and query collections indexed with single terms, after stemming and stopword removal.

Though some of these collections include information other than titles and abstracts, the indexing and retrieval experiments conducted for the present study make use only of the natural language text taken from the title and abstract of each document. Thus, for example, the sections containing keywords and key-phrases in the CACM and INSPEC collections have not been used, and the subject categories assigned to CACM documents have been excluded. The queries are also natural language statements of information need.

1.3. Motivation for Phrase Indexing

1.3.1. Term Specificity

The objective of document retrieval is to identify documents that are related to a particular topic. Doing this effectively requires that the system be capable of distinguishing documents that are relevant to a query from those that are not. Since a document is represented by a set of content indicators, the characteristics of these content indicators determine the degree to which relevant and non-relevant documents can be successfully distinguished.

It would be ideal if one could compile a vocabulary of descriptors having characteristics such that the descriptors would do the best possible job of distinguishing relevant from non-relevant documents. However, since relevance has to do with the relationship between a document collection and a particular query, it would be difficult, if not impossible, to identify such a vocabulary of descriptors that would be ideal for all possible queries. A more realistic goal would be to compile a vocabulary of descriptors that effectively distinguish one document from another within a collection. This objective is more easily attainable, since it has to do with the characteristics of the descriptors that represent the documents of a collection, rather than the more complex relationship of relevance between a document and a query.

In considering the quality of an indexing vocabulary, an important characteristic is the *specificity* of the descriptors that make up the vocabulary. Single words (or word stems) are not necessarily ideal content indicators. This is due, at least in part, to the fact that words vary widely in specificity. Highly specific words identify a narrow range of concepts, whereas very general words may be associated with a broad range of concepts. For purposes of document retrieval, neither very specific nor very general descriptors are ideal, because they retrieve either too few or too many documents. Descriptors of moderate specificity are most desirable because they retrieve a sufficient number of documents to be useful without burdening the user with a large number of documents, many of which will not be of interest.

The quality of an indexing vocabulary can be improved by reducing the variation in specificity of the descriptors that make up the vocabulary. That is, descriptors with excessively low and excessively high specificity could be modified in a manner that yields primarily descriptors having moderate specificity. But in order to do this, it is necessary to first have a means of characterizing specificity in a concrete way, and a means of determining the level of specificity of each descriptor in the vocabulary. The term discrimination model provides a method of classifying descriptors in this way.

The *term discrimination model* relates term specificity to the idea of the *discrimination value* of a term.[7] The discrimination value of a term is an indication of the effect the term has on the average density of the vector space that represents the document collection. A dense vector space is one in which the documents share a relatively large number of descriptors and therefore tend to cluster together in the document space. In a sparse vector space, documents share few descriptors, and therefore tend to be separated from one another in the document space.

When used as a descriptor, a particular term could have one of three possible effects on the average density of the document space. It could increase or decrease the average density, or leave the density unchanged. The discrimination value of a term is defined to be negative if it increases the average density, since the term brings documents closer together in the space and makes it more difficult to distinguish one document from another. Such terms are called negative, or poor discriminators. The discrimination value of a term is positive if the term decreases the average density, since such a term disperses the documents and thus makes it easier to distinguish one document from another. These terms are called positive, or good discriminators. A discrimination value that is near zero indicates that the term has a negligi-

[7] The *discrimination value* or *term discrimination* model was initially proposed as a term weighting method (Salton and Yang 1973). General discussions of the model can be found in Salton (1975a:443-461), Salton (1975b:8-10, 41-55), and Salton and McGill (1983:66-71, 84-87, 104-110). The application of the model to construction of thesaurus classes and phrases is treated in Salton, Yang, and Yu (1974, 1975) and Salton and Wong (1976). Yu, Salton, and Siu (1978) present proofs demonstrating that, under certain conditions, application of certain procedures based on the discrimination value model must yield improvements in retrieval effectiveness.

ble effect on the average density of the document space and thus has little effect in distinguishing documents throughout the collection. These terms may be called indifferent, or non-discriminators.

An important insight provided by the term discrimination model is the relationship between the discrimination value of a term and its document frequency, where the document frequency of term t, df_t, is defined as the number of documents in which term t occurs at least once. In general, poor discriminators have high document frequencies, good discriminators have moderate document frequencies, and indifferent discriminators have low document frequencies.

Given this relationship, discrimination value and document frequency can be related to the idea of specificity. Poor discriminators tend to occur in a large proportion of the documents of a collection, and thus tend to have low specificity; these are likely to be very general terms. Non-discriminators occur in very few documents, and have excessively high specificity; these are likely to be very narrow terms. Good discriminators occur in a moderate number of documents, and are likely to have a moderate level of specificity. The relationships among discrimination value, document frequency, specificity, and descriptor quality are summarized in Table 1.2.

Since the discrimination value and document frequency of a term can be determined directly from the distribution of a term in the documents of a collection, the above stated relationship between specificity on the one hand, and discrimination value and document frequency on the other provides an objective means of classifying descriptors with regard to specificity. Once this is done, the overall quality of the indexing vocabulary can be improved by reducing the variation in the specificity of descriptors. The objective is to transform poor and indifferent discriminators into good discriminators. That is, the overly specific non-discriminators must be made more general, and the excessively general poor discriminators must be made more specific. This can be done by constructing two types of complex content indicators: thesaurus classes and phrases.

Descriptor Characteristic	Descriptor Quality (power of discrimination)		
	Good	Poor	Indifferent
Discrimination Value	> 0	< 0	$\simeq 0$
Document Frequency	moderate	high	low
Specificity	moderate	low	high

TABLE 1.2. Relationship of descriptor quality to discrimination value, document frequency, and specificity.

Thesaurus classes can be formed by combining sets of low document frequency non-discriminators that are related in meaning into groups. The resulting classes will be more general than any of their highly specific members, and will also have higher document frequencies. Phrases can be constructed by grouping pairs (or larger combinations) of high document frequency poor discriminators together. The resulting phrases will have lower document frequencies, and will be more specific than their high document frequency components. The nature of thesaurus classes and phrases is discussed further in the following section on term relationships.

An important part of the motivation for phrase indexing is therefore to improve the quality of the indexing vocabulary by reducing variation in the specificity of descriptors. The objective is to create phrases having moderate specificity by constructing complex descriptors that contain terms of low specificity. Section 1.4 discusses several approaches to phrase construction.

It should be noted that the term discrimination model treats the issues of term specificity and the quality of an indexing vocabulary entirely from the perspective of the distributional characteristics of terms in the document collection. These distributional characteristics are, in fact, important in determining the quality of an indexing vocabulary for purposes of document retrieval. However, the notion of term specificity can also be viewed from a semantic perspective. This is the point of view presented in the following section.

1.3.2. Term Relationships

An important objective of phrase indexing is to construct content indicators having an appropriate level of specificity. The construction of good quality phrases, however, depends on more than just the specificity of the component terms of a phrase. In particular, in order to construct semantically appropriate phrases, it is necessary to identify pairs (or larger groups) of words that enter into a particular type of relationship with one another. The purpose of this section is to define the kinds of relationships that are to be treated as phrases, and equally as important, to distinguish them from other kinds of relationships that are useful in content analysis. To this end, two general types of relationship are defined: the *thesaurus relationship*, and the *phrase relationship*. It is important to clearly distinguish these two relationships, since in order to get good retrieval results, they should be handled in different ways in indexing. A brief overview of how these relationships have been used in content analysis is given, together with a discussion of manual and automatic methods used to identify term relationships. This section concludes by pointing out that statistical term associations do not distinguish between the phrase relationship and the thesaurus relationship, and thus that content analysis methods based on them cannot make the best possible use of term relationships.

There is an important relationship between the notion of term specificity as discussed in section 1.3.1 and the term relationships defined in this section. In the term discrimination model, term specificity is used to characterize terms from the perspective of how terms are distributed throughout the documents of a collection. Specificity is important because of the effects it has on the density of the document space. From this point of view, term relationships are of interest because they can be used to alter the specificity of the terms used as document descriptors. The term discrimination model treats term specificity exclusively as a distributional matter. In contrast, this section discusses term relationships from a semantic perspective. The thesaurus relationship and the phrase relationship are different, though very general, semantic relationships. Their connection with term specificity is direct: thesaurus relationships can be used to create descriptors having lower specificity (greater generality), and phrase relationships can be used to create descriptors having higher specificity.

Automatic methods of content analysis are based on the idea of extracting words from the text of a document, and using the resulting set of words as a representation of the document's content. Many useful content indicators can in fact be identified in this way. However, there are two significant drawbacks to this basic strategy.

First, if only words from the text of a document are used to represent that document, then the document can be retrieved only if a query contains some subset of exactly those words. This means that it is the user's responsibility to include all possible appropriate terms in his query, since he has no way of knowing what terms have been actually used to index the documents of the collection. For example, some documents having to do with coniferous trees might contain the term *conifer*; others might contain the term *evergreen*. In order to retrieve all relevant documents, the user would have to be aware of this fact, and use both terms in his query. In this case the relationship is obvious, and a knowledgeable user would probably have no difficulty in including both terms. In general, however, users are unlikely to think of all appropriate terms. The result is that some relevant documents may not be retrieved, so recall will suffer.

Second, since each document is represented by a set of disassociated words, no indication of syntactic or semantic relationships among words is preserved. A problem may arise if, for example, a user's query contains a phrase like *computer science*, which would be represented in the formal query as two disassociated words *computer* and *science*. Though these query terms have a linguistically valid phrase as their source, they will match corresponding terms in documents regardless of whether they have a linguistically valid phrase as source, or come from separate phrases like *computer technology* and *library science*. With inappropriate matches of this kind, non-relevant documents are likely to be retrieved, resulting in a loss of precision.

The first of these problems can be alleviated by identifying and properly handling terms that enter into a thesaurus relationship with one another. The second can be alleviated by identifying and properly handling terms that enter into a phrase relationship with one another.

The thesaurus relationship includes semantic relationships such as synonymy, hyponymy (inclusion), and instantiation. For purposes of indexing, thesaurus relationships are typically handled by constructing a thesaurus containing a number of thesaurus classes. Each class consists of a group of terms that enter into a thesaurus relationship with one another. The most restricted form of thesaurus consists of groups of synonyms, or at least very closely related words. The words *production* and *manufacture*, for example, could be members of a class. In indexing, then, if a document contains either of these terms, it would be assigned a descriptor that represents the class as a whole, rather than just one of the terms. Queries are treated in the same way, so that a query containing *production of automobiles* would match documents originally containing either *production of automobiles* or *manufacture of automobiles*. An alternative to assigning a descriptor that represents the class as a whole is to simply add to a query all the members of a thesaurus class represented in the original query. Both of these approaches have the effect of broadening the query by increasing the possibilities for matches between queries and documents. This is therefore a recall enhancing device.

While some manually constructed thesauruses may be restricted to well-defined synonym classes, thesaurus classes are typically much more loosely defined. This is especially true of term groupings derived by automatic means. The work of Jones and Sinclair (1974:38-42) illustrates the variety of relationships that hold between pairs of words that are associated statistically; further examples can be found in Salton (1968:131, Table 4-2). Because of the loosely defined character of thesaurus classes, in practice, the thesaurus relationship includes virtually any kind of relationship that holds between terms that are related due to the fact that they refer to different aspects of a common concept or domain. The thesaurus concept has also been extended to include hierarchical relationships among thesaurus classes. (A general discussion of the construction and application of thesauruses can be found in Salton (1975a:461-471).)

The relationships among members of a thesaurus class are due to the inherent meanings of the words involved, rather than to the grammatical structure of the text in which they occur. The thesaurus relationship can be viewed as a type of paradigmatic relation (Lyons 1968:73-74; Gardin 1973:147).

The phrase relationship can be defined as a relationship of modification or specification. Some examples are: *text analysis*, *structural linguistics*, and *computer science*. In each case, the first element of the phrase modifies the second, so that the phrase as a whole refers to a more specific concept. It is useful to extend the phrase relationship to include not just nouns and their modifiers, but also relationships that hold between verbs and their arguments. This makes it possible, for example, to recognize that the sentence in (1.6) contains a phrase that is essentially the same semantically as the noun phrase *text analysis*.

(1.6) The system *analyzes text* automatically.

In indexing, phrase relationships can best be handled by identifying terms that are related in the appropriate way, and then assigning a phrase descriptor that represents the phrase as a whole, rather than (or perhaps in addition to) the less specific, individual descriptors that represent the elements of the phrase.

Unlike the thesaurus relationship, the phrase relationship is not primarily dependent on the inherent meanings of the words involved, but on the grammatical structure of the text in which they occur. The phrase relationship is thus a syntagmatic relation (Lyons 1968:73-74; Gardin 1973:147).

In summary, the proper treatment of thesaurus and phrase relationships is as follows:

(1) If terms A and B are members of the same thesaurus class C, then if A occurs in the text of a document, assign both A and B as descriptors. Alternatively, assign descriptor C, representing the class as a whole. Similarly, if B occurs in a document, assign both A and B, or alternatively C, as descriptors. Schematically,

if (A **or** B), assign (A **and** B) **or** C.

(2) If terms A and B occur in a document, and enter into a relationship of modification or specification with one another, then assign phrase AB as a descriptor. Schematically,

if (A **and** B), assign AB.

Treating thesaurus relationships as in (1) results in a broader, more general content representation that enhances recall. In contrast, treating phrase relationships according to (2) results in a narrower, more specific representation that enhances precision.

Manually constructed thesauruses are most often compiled by subject experts who use their familiarity with the literature of a subject area to identify groups of related terms. Likewise, phrase dictionaries can be compiled by gathering phrases that refer to important concepts in a particular subject area, and that occur commonly in documents dealing with that area. Computational aids are sometimes used to facilitate the construction of both thesauruses and phrase dictionaries (Salton 1968:25-30, 1975a:461-471).

Substantial effort has also been directed toward developing fully automatic methods of identifying related terms from the text of documents. These methods make use of measures of term association (correlation) based on the frequency with which pairs of terms cooccur in the documents of a collection (Doyle 1961, 1962; Stiles 1961; Giuliano and Jones 1963; Giuliano 1965; Lesk 1969; Salton 1972b).

Some researchers have claimed that it is possible to identify different kinds of term relationships automatically. Giuliano and Jones (Giuliano and Jones 1963; Giuliano 1965), for example, say that it is possible to distinguish what they call "contiguity association" from "synonymy association" by generating first and second order term associations. Bruandet's (1987) method of recognizing associated terms identifies relationships that are similar to these second order terms associations. It appears that little work has been done, however, to determine the influence that associations of this kind have on retrieval effectiveness.

Since the groups of associated terms generated by these associative methods are determined by the cooccurrence characteristics of words in text, it is necessarily the case that some groups will represent both thesaurus relationships and phrase relationships. In spite of this fact, they are typically used as if only thesaurus relationships were involved. That is, in practice, such term groupings are most often used for query expansion (Giuliano and Jones 1963; Stiles 1961; Lesk 1969; Salton and McGill 1983:78-84). Though the overall retrieval strategy differs significantly, the term dependency models of the more recent probabilistic retrieval methods use term associations for query expansion in essentially the same way (Salton, Buckley, and Yu 1983; Yu et al. 1983).

Because the associative methods based on cooccurrence characteristics of terms cannot differentiate phrase relationships and thesaurus relationships, the proper treatment of these relationships cannot be consistently maintained.

1.4. Construction of Phrase Descriptors

Two important considerations in constructing phrase descriptors are term specificity and term relationships. This section presents an overview of methods that can be applied in an effort to construct phrase descriptors that have an acceptable level of specificity, and that contain words related in appropriate ways. Phrase construction involves two processes: phrase identification and phrase normalization.

1.4.1. Phrase Identification

Phrase identification is the process of identifying in the text of documents and queries groups of words that can be combined to form phrase descriptors. This selection procedure may take into consideration a variety of characteristics of terms (words or word stems) and the texts in which they occur. These characteristics include: (1) the frequency of occurrence of words in a document collection, (2) the proximity of words in text, (3) the syntactic structure of texts, and (4) semantics.

Information about the frequency of terms can be incorporated into the phrase identification process in various ways. One approach is to use the document frequency of individual terms to identify those terms that should be included as elements of phrase descriptors. This is the basis of the phrase indexing method of the term discrimination model (Salton, Yang, and Yu 1974, 1975; Salton and Wong 1976; see also section 1.3.1). The document frequency of term t, df_t, is the number of documents in the collection in which term t occurs. Terms with a high document frequency are likely to be very general terms that could be improved by combining them with other terms to form phrases with more specific meanings, and lower document frequencies. For example, terms like *system*, *computer*, and *programming* would have high document frequencies in a collection of computer science documents. By constructing phrases that contain these terms, for example, *information system*, *computer programming*, and *programming language*, more specific descriptors can be introduced into the indexing vocabulary.

Another approach is to consider the frequency of the phrase itself rather than the frequencies of its elements. The idea here is that a phrase that occurs frequently in a collection is more likely to be a meaningful, semantically appropriate phrase, than one that has a very low frequency of occurrence. This approach has been used with some success by several researchers (see for example, Steinacker 1973, 1974; Olney, Lam and Yearwood 1976; Neufeld, Graham, and Mazella 1974).

A further refinement of the use of frequency information is to take into consideration the cooccurrence characteristics of pairs or larger groups of terms. Terms that cooccur in a specified unit of text at a frequency higher than would be expected given their individual frequencies are more likely to be semantically valid phrases than are pairs of terms with lower cooccurrence frequencies. The statistically oriented term association methods described in section 1.3.2 are based on this idea, as are the term dependency models used in probabilistic retrieval environments (Salton, Buckley, and Yu 1983; Yu et al. 1983).

The objective of using information about the frequency of phrases and the cooccurrence frequency of terms is to increase the chances that the terms included in a phrase descriptor form a semantically valid phrase, rather than just a random association of terms. Another approach to attaining this objective is to construct phrases only from terms that occur in close proximity to one another in texts. Phrases formed from terms that are adjacent in a text, or that are separated by only one or two other terms are more likely to be good phrases than if the component terms were more widely separated. In addition to simple proximity, other cooccurrence requirements may also be specified. For example, it may be required that terms occur in the same sentence, or may not be separated by certain kinds of punctuation.

Information about the proximity of terms is typically used in conjunction with frequency characteristics. That is, in order to be used as a phrase descriptor, a pair of terms would have to meet certain proximity requirements in addition to having specified frequency characteristics. This is the case, for example, in the discrimination value phrase construction method, and the approaches used by Steinacker, Olney, and Neufeld as cited above. In addition to proximity and cooccurrence criteria, the method for identifying term associations developed by Bruandet (1987) also incorporates information about word classes.

The primary advantages of using simple frequency and proximity information for phrase identification, is that these methods are easy to implement, are not excessively demanding on computing resources, and do not require special adjustments in order to be applied to a variety of different document collections.

A further refinement in the process of phrase identification is to take into consideration the syntactic structure of the text that is being indexed. By making use of information about the syntactic structure of text, it is possible to avoid constructing phrases from groups of terms that are not related in appropriate ways, even though they may occur in close proximity. For example, a phrase identification procedure based on word stems and proximity information would construct the phrase *comput sci* from the common phrase *computer science*, as well as from the phrase in (1.7) even though the two sources do not refer to the same concept.

(1.7) the use of *computers* in *science* and technology

Another, perhaps more valuable benefit of syntactic information is that it can be used to identify terms that are related syntactically in an appropriate way for phrase construction, but do not occur in close proximity to one another. This situation is illustrated by the noun phrase in (1.8).

(1.8) preparation and evaluation of abstracts and extracts

Knowledge of the syntactic structure of this phrase makes it possible to identify *abstract preparation*, *abstract evaluation*, *extract preparation*, and *extract evaluation* as phrase descriptors, while at the same time avoiding the construction of inappropriate phrases like *preparation evaluation* and *abstract extract*.

A number of researchers have made efforts to use syntactic information for purposes of content analysis in document retrieval. These range from relatively simple segmentation and pattern matching techniques based on word classes (Baxendale 1958, 1961; Klingbiel 1973a, 1973b; Dillon and Gray 1983; Dillon and McDonald 1983; Aladesulu 1985), to more general partial syntactic analysis procedures (Vladutz 1983; Vladutz and Garfield 1979; Melton 1966; Earl 1970, 1972; Hillman 1968, 1973; Hillman and Kasarda 1969), and finally to systems capable of complete syntactic analysis (Salton 1966;

Young 1973; Metzler et al. 1984).

The refinements in identifying relationships among terms that are offered by syntactic information provide the potential to significantly improve the process of phrase identification in comparison to the simpler methods based on frequency and proximity considerations. A further benefit of using an approach to syntactic analysis that does not depend on detailed semantic information is that the analysis procedure is not restricted to texts of a single domain of discourse. Chapter 3 gives further consideration to the application of syntax to the problem of phrase identification.

Although syntax does make it possible to identify relationships among terms that cannot be accurately recognized by simpler means, there are also problems in identifying term relationships that cannot be solved by syntax alone. In cases where the syntactic structure of a phrase or sentence is ambiguous, semantic information must be brought into play. Complex nominal constructions illustrate this problem. For example, in the noun phrase in (1.9), syntactic information is not sufficient to determine whether *frequency* modifies *transistor* or *oscillator*.

(1.9) high frequency transistor oscillator

Similarly, *high* could modify any of the three words to its right. In order to correctly determine the structure of this phrase, semantic information must by provided to indicate that *frequency* is a possible modifier of *oscillator* but not of *transistor*, and that *high* is a common modifier of *frequency* but not of *transistor* or *oscillator*. In many cases, quite detailed domain-specific semantic information may be required to resolve ambiguities of this kind. However, some benefit can be derived from more general semantic information that is not tied to any specific domain. Sparck Jones and Tait (1984a, 1984b) have investigated the use of semantic information of this kind.

1.4.2. Phrase Normalization

In addition to identifying useful phrases in the text of documents, it is also desirable to recognize groups of phrases that differ in form but that are similar enough semantically to be represented by a single phrase descriptor. For example, it is beneficial to recognize that the text phrases *information retrieval* and *retrieval of information* are essentially identical in meaning and therefore can be represented by the same phrase descriptor. This is the objective of phrase normalization.

In the simplest case, normalization is accomplished by deleting function words and ignoring the order of words in phrases. This has the desired effect for pairs of phrases like those in (1.10).

(1.10) *information retrieval* ~ *retrieval of information*
book review ~ *review of books*

However, some incorrect normalization also results from this method. For example, in (1.11) the adjacent words *system* and *operating* are identified as a phrase and represented by the same phrase descriptor as *operating system*, even though they do not refer to the same concept.

(1.11) An online *system operating* as part of a normal batch system for the CDC6600 computer is described.

A similar problem occurs with pairs of phrases like *science library* and *library science*. By ignoring the order of phrase elements, an important semantic distinction is lost. This method of normalization has been used in the phrase indexing experiments based on the term discrimination model (Salton, Yang, and Yu 1975; Salton and Wong 1976), as well as the syntax-based procedures of Dillon and Gray (1983).

A similar normalizing effect can be accomplished by the approximate phrase matching procedure of Paice and Aragón-Ramírez (1985). Their objective is to determine the degree of similarity of pairs of phrases such as *binary tree* and *binary search tree*. Their procedure is to establish a mapping between the individual words in the two strings, and then calculate a similarity value that is a function of (a) the number of shared elements, (b) the total number of elements, and (c) the order of elements. In a related approach, Rodger Knaus (1983) uses probabilistic considerations to map natural language phrases into a predetermined vocabulary of standardized phrases.

These strategies do accomplish some useful normalization, but at the same time they frequently yield inaccurate representations. This is because the meaningful relationships that hold between the elements of a phrase are not taken into consideration.

By making use of information about the syntactic structure of phrases, many inaccuracies introduced by the simpler approaches to normalization can be avoided. For example, syntactic analysis of the phrases *information retrieval* and *retrieval of information* provides the information that in both cases *retrieval* is the head of the construction and that *information* is the modifier. Given this information, it is clear that both phrases can accurately be represented by the phrase descriptor *information retrieval.* In contrast, given a syntactic analysis of the phrases *library science* and *science library,* it can be established that in the first case *science* is the head of the construction and *library* is its modifier, whereas the reverse is true for *science library.* This structural information makes it possible to avoid incorrectly representing this pair of phrases by the same phrase descriptor. Methods of normalization based on syntactic structure are discussed further in chapter 3.

CHAPTER 2

NON-SYNTACTIC PHRASE INDEXING

2.1. Introduction

The approach to phrase construction presented in this chapter is based on the ideas of term specificity and discrimination value as discussed in section 1.3.1. The method is considered non-syntactic because only the frequency and cooccurrence characteristics of terms are taken into consideration in constructing phrases. The objective of the chapter is to establish the level of effectiveness that can be achieved using this simple, non-syntactic phrase indexing strategy, and to examine various problems related to the quality of the phrase descriptors constructed using this procedure.

2.2. Non-Syntactic Phrase Indexing Method

The phrase indexing procedure is described by first presenting a general overview, and then defining and explaining the purpose of the parameters on which the procedure is based. Finally, the procedure is illustrated by applying it to a sample document.

2.2.1. Overview and Definition of Parameters

This phrase construction method is based on the one proposed by Salton, Yang, and Yu (1975). It has been generalized, however, to make it possible to test some extensions to their original method. The procedure is controlled by seven parameters that incorporate the notion of term specificity and the cooccurrence characteristics of terms into the phrase contruction process.

The outline below, together with the parameter definitions that follow it, constitutes a complete description of the phrase indexing process. The process is illustrated by the example in section 2.2.2.

(1) Non-Syntactic Phrase Indexing Procedure

(a) Construct a dictionary of phrases, if desired.

(b) Apply the phrase construction procedure to documents and queries to construct candidate phrases.

(c) Assign phrase descriptors to documents and queries.

(d) Assign single term descriptors to documents and queries.

(2) Phrase Dictionary Construction Procedure

(a) Select a corpus of text from which phrases are to be selected.

(b) Apply the phrase construction procedure to this corpus to get a set of candidate phrases.

(c) Apply the phrase selection criteria (if any) to candidate phrases. Each candidate phrase that meets these requirements goes into the phrase dictionary.

(3) Phrase Construction Procedure

(a) Identify terms that are acceptable as phrase elements. There are two kinds of phrase elements: *phrase heads* and *phrase components*.

(b) For each phrase head in a specified domain of cooccurrence, construct a candidate phrase containing the phrase head and cooccurring terms such that the phrase length, domain of cooccurrence, and proximity requirements are maintained. A phrase may not contain two identical elements.

(4) Assignment of Phrase Descriptors

(a) If a phrase dictionary is being used, a candidate phrase is assigned as a phrase descriptor only if it is in the phrase dictionary.

(b) If a phrase dictionary is not being used, all candidate phrases are assigned as phrase descriptors.

(c) A phrase is assigned as a descriptor to a query only if the phrase also occurs in at least one document.

(5) Assignment of Single Term Descriptors

(a) Terms not included in phrases are assigned as single term descriptors, provided that the selection criteria for single terms are met.

(b) Terms included in phrases are assigned as single term descriptors, provided that the selection criteria for phrase elements are met.

(c) Different selection criteria can be specified separately for single terms not included in phrases, phrase heads, and phrase components.

(d) A single term is assigned as a descriptor to a query only if the term also occurs in at least one document.

Parameter Definitions:

domain: The domain of cooccurrence of phrase elements. The elements of a phrase must cooccur in a specified unit of text. This domain of cooccurrence is specified by the *domain* parameter. Possible domains of cooccurrence are the document (or query), the paragraph, and the sentence. As the domain of cooccurrence becomes more restricted, the total number of phrases constructed is reduced. In addition, terms that occur in a restricted domain are more likely to form a meaningful phrase than those that occur in a less restricted domain. For example, adjacent terms that straddle a sentence boundary may be less likely to form a meaningful phrase than adjacent terms within a single sentence.

proximity: The relative location of phrase elements. The domain parameter specifies the unit of text within which phrase elements must cooccur. The *proximity* parameter specifies the allowable distance between phrase elements that cooccur within a given domain. Like the domain parameter, the proximity parameter is used as a means of increasing the likelihood that the elements of a phrase are related in a meaningful way, rather than being just a random collocation. That is, words that occur in close proximity to one another in a document or query text are more likely to form a meaningful phrase than words that are widely separated. Proximity is defined in terms of the distance between words. Adjacent words are at a distance of one from one another; words separated by one intervening word are at distance two, etc. The distance between words is measured after stopwords have been removed.

df-phrase: Document frequency threshold for phrases. The parameter *df-phrase* has been included in this phrase indexing model in order to test two hypotheses. The first hypothesis is that phrases with low document frequencies may have a detrimental effect on retrieval effectiveness, since such phrases are more likely to be random collocations rather than meaningful phrases. Low document frequency phrases are excluded by selecting a threshold, $df\text{-}phrase_{min}$, and then assigning phrase p as a descriptor only if $df_p \geq df\text{-}phrase_{min}$. The second hypothesis is that phrases with very high document frequencies are likely to be detrimental to retrieval effectiveness since the elements of high document frequency phrases are typically high document frequency single terms.[1] The effect of these phrases could be that matches on high document frequency phrases reinforce the effect of matches on their general, high document frequency elements. This would be expected to result in a loss of precision. High document frequency phrases are excluded by selecting a threshold, $df\text{-}phrase_{max}$, and then assigning phrase p as a descriptor only if $df_p < df\text{-}phrase_{max}$.

This parameter also has the effect of placing a threshold on the cooccurrence frequency of the elements of phrases. That is, if $df_p = 10$, then the elements of phrase p have a cooccurrence frequency in the document collection of at least ten. This parameter is typically used as a criterion for selecting phrases to be included in the phrase dictionary, and thus for selecting the set of phrases that can be assigned as content indicators.

df-head: Document frequency threshold for phrase heads. Within the framework of the term discrimination model, a primary objective of phrase indexing is to construct phrases that contain poor discriminators in order to produce phrases having better discrimination values than the individual terms used to construct them. In order to assure that phrase indexing will have this effect, it is required that all phrases contain at least one high

[1] Here, "high document frequency" refers to the document frequency range for phrases, not single terms. In all collections, the highest document frequency for phrase descriptors is much lower than for single terms.

document frequency, poor discriminator. This element of the phrase is called the *phrase head.* The parameter *df-head* is a document frequency threshold used for identifying phrase heads. Term t is acceptable for use as a phrase head if $df_t \geq df\text{-}head$, where df_t is the document frequency of term t.

df-comp: Document frequency threshold for phrase components. In addition to the obligatory phrase head, each phrase contains another term which may have a lower document frequency than a phrase head. This element of the phrase is called the *phrase component.* The document frequency threshold *df-comp* is used to identify phrase components. Term t is acceptable as a phrase component if $df_t \geq df\text{-}comp$.

In addition to controlling the document frequency of phrase elements that are not phrase heads, df-comp also makes it possible to avoid constructing phrases that have very low document frequencies. If a phrase head is combined with a term having a very low document frequency, the resulting phrase will have a document frequency that is at least as low as, but often lower than, the document frequency of the low frequency element. Thus by using somewhat higher values for df-comp, the number of very low document frequency phrases can be reduced.

df-st: Document frequency threshold for single term descriptors. In section 1.3 it was explained that the term discrimination model provides a basis for improving the quality of an indexing vocabulary. One aspect of this process is the construction of phrases containing high document frequency, poor discriminators. By constructing phrases containing high document frequency terms, new descriptors are produced that have lower document frequencies and better discrimination values than their high document frequency elements. The question remains, however, whether the single terms that are included in phrases should be replaced by the phrase descriptors, or whether the single terms should be kept as single term descriptors along with the phrases.

The parameter *df-st* is a document frequency threshold that is used as a criterion for selecting single terms to be assigned as single term descriptors. Term t is acceptable as a single term descriptor if $df_t < df\text{-}st$. This threshold can be used to assure that high document frequency, poor discriminators are not assigned as single term descriptors. This selection criterion can be appled to all single terms, or just to those single terms that are actually used as elements of phrase descriptors.

length: The number of elements in a phrase. The length parameter simply specifies the maximum number of terms a phrase may contain. All phrases used in this study have a length of two. This length has been used in order to control the number of phrase types and phrase tokens identified in document and query texts. As the value of the length parameter is increased, the number of phrase types increases dramatically, while the number of tokens corresponding to each type becomes very small. The overall result is that a very large number of distinct phrases may be assigned as descriptors, but since the frequency of occurrence of most phrases is very low, the vast majority of phrases would have a negligible effect on retrieval performance. An additional consideration is that a greater phrase length tends to increase the number of random collocations that are identified as phrases, since the distance between phrase elements increases as phrase length increases.

2.2.2. Non-syntactic Phrase Indexing Example

Some sample values for the parameters defined above are given in Table 2.1.

domain	proximity	df-phrase	df-head	df-comp	length
sentence	1	1	55	1	2

TABLE 2.1. Sample phrase indexing parameter values applied to CISI document 71.

Using these values, the details of the phrase indexing procedure can be clarified by describing its application to the title of document 71 from the CISI collection.

Using 'sentence' as the value of the domain parameter specifies that phrase elements must cooccur in the same sentence, and a proximity of one specifies that phrase elements must be adjacent after removal of stopwords. A value of one for df-phrase places no restrictions on the document frequency of phrases. A value of 55 for df-head assures that all phrases will contain at least one term having a document frequency of at least 55. Using a value of one for df-comp places no restrictions on the document frequency of the other element of a phrase. Since the minimum document frequency of a descriptor is one, any term can be combined with a phrase head to form a phrase. Finally, the length parameter specifies that a phrase may contain only two elements.

The indexing procedure begins by identifying individual word tokens in the text (Figure 2.1) removing stopwords, and performing a stemming operation.[2] At the same time, section, paragraph, and sentence boundaries are recognized. The result of this step is illustrated in Figure 2.2.

Word-Word Associations in Document Retrieval Systems

FIGURE 2.1. Original title of CISI document 71 (Lesk 1969).

Token	Desc. Type	Doc. No.	Para. No.	Sen. No.	Tkn. No.	Doc. Freq.	Phrase Head	Phrase Comp.
word	0	71	1	1	1	99	YES	YES
word	0	71	1	1	2	99	YES	YES
associ	0	71	1	1	3	23	no	YES
docu	0	71	1	1	5	247	YES	YES
retrief	0	71	1	1	6	296	YES	YES
system	0	71	1	1	7	535	YES	YES

FIGURE 2.2. Input to phrase construction procedure for CISI document 71.

This information is used as the input to the phrase construction procedure. The columns labeled 'Phrase Head' and 'Phrase Comp.' in Figure 2.2 indicate the status of each token with regard to its acceptability as a phrase head and as a phrase component, as determined by the document frequency of each token and the values of the df-head and df-comp parameters. Phrase con-

[2] The stemming algorithm is based on the work of Lovins (1968).

struction proceeds by combining pairs of adjacent tokens that meet the document frequency requirements for phrase heads and phrase components. For example, in Figure 2.2, the token *docu* is acceptable as a phrase head, so it is combined with adjacent tokens *associ* and *retrief* to form phrases *docu associ* and *docu retrief*. Similarly, the tokens *retrief* and *system* are both acceptable as phrase heads (as well as phrase components), and therefore combine to form the phrase *retrief system*. The order of phrase elements is regularized so that a pair of phrases cannot differ by order alone. Also, a phrase descriptor may not be constructed from two identical elements, so *word word* is not assigned as a phrase descriptor, even though the document frequency and proximity requirements for these tokens are met.

Figure 2.3 illustrates the final vector form of document 71, which consists of two subvectors: the single term subvector containing descriptors of type 0, and the phrase subvector containing descriptors of type 1.[3]

Document Number	Descriptor Number	Weight	Descriptor Type	Descriptor
71	26546	0.5706	0	associ
71	26850	0.2194	0	retrief
71	34344	0.7399	0	word
71	34406	0.2443	0	docu
71	39899	0.1380	0	system
71	10365	0.1787	1	retrief system
71	17459	0.2318	1	docu retrief
71	21114	0.6553	1	word associ
71	24244	0.4075	1	docu associ

FIGURE 2.3. Final form of vector for CISI document 71.

[3] See the work of Fox (1983a, 1983b) for further discussion of vectors containing multiple concept types.

This phrase indexing procedure has been implemented in C, and is designed to interface easily with the SMART package (Buckley 1985).

2.2.3. Weighting and Similarity Functions

2.2.3.1. Weighting of single term descriptors

The weight assigned to a descriptor in a vector is indicative of the importance of the descriptor as an indicator of document or query content. In order to include information about the relative importance of a term in an individual document or query, the weighting function used in these experiments incorporates the frequency of each term in a given document or query. As an indication of the quality of a descriptor with respect to the document collection as a whole, the inverse document frequency ratio is included. A discussion of these two weighting factors can be found in Sparck Jones (1972) and Salton and Yang (1973). The cosine normalization is used in order to normalize for vector length.

The following expressions define the weighting function. Initially, the weight of term t in vector v is the frequency of t in the document or query represented by v. This is a simple term frequency weight, tf_{tv}. The term frequency weights are normalized by dividing by the maximum term frequency in the vector, max_tf_v, as shown in (2.1).

$$norm\_tf_{tv} = \frac{tf_{tv}}{max\_tf_v} \tag{2.1}$$

The inverse document frequency ratio is incorporated with the definition given in (2.2), where n is the number of documents in the collection, and df_t is the document frequency of t, that is, the number of documents in which term t occurs at least once.

$$tf\_idf_{tv} = norm\_tf_{tv} \cdot \ln \frac{n}{df_t} \tag{2.2}$$

The cosine normalization yields the final weight, w_{tv}, of term t in vector v, as shown in (2.3), where k is the length of vector v.

$$w_{tv} = \frac{tf\_idf_{tv}}{\sqrt{\sum_{i=1}^{k} tf\_idf_{iv}{}^2}} \tag{2.3}$$

The weights defined by expressions (2.1)-(2.3) are used for single term descriptors in collections that are indexed with single terms only, as well as for single term descriptors in collections indexed with both single terms and phrases. In collections indexed with both single terms and phrases, however, normalization is done over the single term subvector only, rather than over the entire vector. Thus the single term subvector for a document (or query) in a collection indexed with single terms and phrases is identical to the vector for the same document (or query) in a collection indexed with single terms only.

2.2.3.2. Weighting of phrase descriptors

The weight of a phrase descriptor is a function of the weights of its elements. If phrase p in vector v is composed of single terms a and b, also in vector v, then the weight, w_{pv}, of phrase p in vector v is given by the expression in (2.4).

$$w_{pv} = \frac{w_{av} + w_{bv}}{2} \tag{2.4}$$

This phrase weight has been chosen for two reasons. First, since the phrase weight is a function of the weights of the phrase elements, it incorporates information about the importance of the elements of the phrase into the phrase weight. Second, it assures that the magnitude of phrase weights does not differ greatly from the magnitude of single term weights.

2.2.3.3. The query-document similarity function

A document or query indexed with both single terms and phrases consists of two subvectors, one containing single term descriptors, and one containing phrase descriptors. In order to calculate the similarity between a query vector and a document vector, a partial similarity is calculated for each subvector, and the overall similarity is then calculated as a weighted sum of the two partial similarities.

Let q represent a query vector consisting of a single term subvector q_s and a phrase subvector q_p; similarly, let d represent a document vector with single term and phrase subvectors d_s and d_p. The simple innerproduct function (2.5) is used as the basic similarity function for a pair of subvectors, for example, q_s and d_s.

$$ip(q_s,\ d_s) = \sum_{i=1}^{k} q_{si} \cdot d_{si} \tag{2.5}$$

Here, k represents the length of subvector s, and q_{si} and d_{si} are the weights of the ith terms in the single term subvectors q_s and d_s.

For single term subvectors to which the cosine normalization has been applied (see (2.3) above), the innerproduct function yields a similarity value equivalent to the cosine similarity function (Salton and Lesk 1968:25) applied to vectors to which the cosine normalization has not been applied.

The overall similarity value for vectors q and d is calculated as a weighted sum of the innerproduct similarity values calculated for the single term and phrase subvectors (see 2.6). Here, c_s and c_p are weights applying to the single term and phrase subvectors, respectively.

$$sim(q,\ d) = (c_s \cdot ip(q_s,\ d_s)) + (c_p \cdot ip(q_p,\ d_p)) \tag{2.6}$$

For the experiments discussed in this chapter, the value 1.0 has been used for both c_s and c_p.

With these weighting and similarity functions, the addition of phrase descriptors to document and query vectors has only a simple additive effect on the overall similarity between a document and query. That is, the partial similarity due to the single term subvector is not altered by the addition of phrase descriptors. The net effect of this strategy for weighting descriptors and calculating similarity values is that phrase descriptors can increase the similarity between a pair of vectors, but cannot reduce the partial similarity due to matches between descriptors in the single term subvectors of the query and document. This would not be the case if the single term and phrase descriptors were not differentiated, and the normalization of expressions (2.1) and (2.3) was done over the entire vector.

2.3. Retrieval Experiments

The objective of phrase indexing is to identify groups of words in text that will enhance retrieval effectiveness when assigned as phrase descriptors to representations of documents and queries. The phrase indexing procedure described above attempts to do this by taking into consideration two simple characteristics of words in text: document frequency and word location. These characteristics are incorporated into the phrase indexing procedure by six parameters: domain, proximity, df-phrase, df-head, df-comp, and length. By varying the values of these parameters, the selectivity of the phrase indexing procedure can be varied greatly. A highly selective procedure results when very restrictive document frequency and cooccurrence characteristics are specified. Such a procedure constructs phrases consisting of pairs of terms with high document frequencies cooccurring in a small domain at close proximity. A highly unselective procedure results when unrestrictive document frequency and cooccurrence characteristics are specified. Such a procedure constructs phrases consisting of essentially any pair of terms cooccurring in the largest possible domain at any proximity.

There is currently no well-motivated basis for selecting parameter values that can be expected to yield good retrieval results for a particular document collection. Thus in order to establish the level of retrieval effectiveness that can be attained with this method of phrase indexing, optimal parameter values must be determined empirically for each experimental document collection. A large number of experiments have been conducted in which the phrase indexing procedure was applied repeatedly, while systematically varying parameter values. This was done for five document collections: CACM, INSPEC, CRAN, MED, and CISI. Basic characteristics of these collections appear in Table 1.1. For each set of parameter values used, a retrieval experiment was done to compare the effectiveness of simple single term indexing to that of phrase indexing. In this way, a set of parameter values that yields optimal retrieval results for this phrase indexing method was established for each collection.

Table 2.2 exhibits the optimal parameter values for each collection, together with retrieval effectiveness figures expressed as percent change in average precision in comparison to simple single term indexing. Table 2.3 contains the corresponding complete recall-precision results.[4] Table 2.4 summarizes the retrieval performance attained when identical phrase construction criteria were applied to all the test collections. These tables show that the responses of the test collections to the phrase indexing procedure were quite variable, both with respect to the level of retrieval effectiveness achieved, and the optimal values of phrase indexing parameters.

---

[4] The average precision figures in Table 2.3 are based on calculations for 21 recall levels. Summary statistics are presented, however, for only ten recall levels, 0.10-1.00.

Collection	Non-syntactic Phrase Indexing Parameters				Avg. Prec. Change	Stat. Signif. Change?
	domain	proximity	df-phrase	df-head		
CACM	doc.	unlimited	< 90 (0.03n)	1	**+22.7%**	yes P < 0.01
INSPEC	doc.	unlimited	< 150 (0.01n)	1	**+11.9%**	yes P < 0.01
CRAN	doc.	unlimited	< 90 (0.06n)	1	+8.9%	yes P < 0.01
MED	sent.	unlimited	≥ 3	3[a]	+4.0%	yes P < 0.01
CISI	sent.	1	< 30 (0.02n)	1	+2.2%	no P > 0.05

TABLE 2.2. Best parameter values and summary of retrieval results. Average precision change is with respect to single term indexing (see Table 2.3); boldface indicates material change. In the df-phrase column, n is collection size; see Table 1.1.

[a] This value for df-head is a by-product of the value for df-phrase; it is not an independently imposed restriction.

With regard to retrieval effectiveness, a statistically significant increase was attained for CACM, INSPEC, CRAN, and MED, as indicated by their changes in average precision.[5] Of these four, however, only CACM and INSPEC show an increase that can be characterized as "material" according to the criteria suggested by Sparck Jones (1974:397). CISI exhibits a slight increase in average precision, which is neither statistically significant nor material.

[5] The significance test used was the Wilcoxon signed rank test for paired observations.

Recall Level	Precision					
	CACM		INSPEC		CRAN	
	Single Terms	Phrases	Single Terms	Phrases	Single Terms	Phrases
0.10	0.5086	0.6489	0.5261	0.6084	0.7526	0.8001
0.20	0.4343	0.5335	0.4181	0.4923	0.6187	0.6704
0.30	0.3672	0.4542	0.3412	0.3893	0.5184	0.5659
0.40	0.2972	0.3569	0.2781	0.3090	0.4282	0.4732
0.50	0.2398	0.2971	0.2283	0.2488	0.3714	0.4116
0.60	0.1912	0.2416	0.1777	0.1900	0.2952	0.3240
0.70	0.1462	0.1719	0.1360	0.1380	0.2301	0.2452
0.80	0.1086	0.1261	0.0936	0.0942	0.1839	0.2001
0.90	0.0711	0.0742	0.0484	0.0527	0.1313	0.1474
1.00	0.0610	0.0615	0.0179	0.0199	0.1175	0.1307
Avg Prec	0.2604	0.3195	0.2459	0.2750	0.3852	0.4194
% Change		22.7		11.9		8.9

(a)

Recall Level	Precision			
	MED		CISI	
	Single Terms	Phrases	Single Terms	Phrases
0.10	0.8036	0.8512	0.4919	0.4947
0.20	0.7258	0.7843	0.4032	0.4026
0.30	0.6742	0.7222	0.3118	0.3285
0.40	0.6317	0.6430	0.2624	0.2712
0.50	0.5447	0.5570	0.2320	0.2330
0.60	0.4728	0.4818	0.1901	0.1982
0.70	0.4082	0.4175	0.1504	0.1556
0.80	0.3501	0.3536	0.1119	0.1131
0.90	0.2057	0.2127	0.0739	0.0811
1.00	0.0888	0.0970	0.0521	0.0582
Avg Prec	0.5378	0.5595	0.2450	0.2503
% Change		4.0		2.2

(b)

TABLE 2.3. Average precision at 10 recall levels for single term and phrase indexing.

Proximity	CACM	INSPEC	CRAN	MED	CISI
Domain: Document					
unlimited	**0.3128** **+20.1%**	**0.2652** **+7.9%**	**0.4169** **+8.2%**	0.5501 +2.3%	0.2167 -11.5%
10	0.3065 +17.7%	0.2591 +5.4%	0.4111 +6.7%	0.5503 +2.3%	0.2261 -7.7%
5	0.2987 +14.7%	0.2617 +6.4%	0.4119 +6.9%	0.5523 +2.7%	0.2320 -5.3%
1	0.2803 +7.6%	0.2546 +3.5%	0.3989 +3.6%	0.5429 +0.9%	0.2396 -2.2%
Domain: Sentence					
unlimited	0.3025 +16.2%	0.2534 +3.0%	0.4105 +6.6%	**0.5555** **+3.3%**	0.2326 -5.0%
10	0.3018 +15.9%	0.2565 +4.3%	0.4126 +7.1%	0.5519 +2.6%	0.2317 -5.4%
5	0.2956 +13.5%	0.2618 +6.5%	0.4082 +6.0%	0.5525 +2.7%	0.2323 -5.2%
1	0.2808 +7.9%	0.2545 +3.5%	0.3991 +3.6%	0.5435 +1.1%	**0.2406** **-1.8%**

TABLE 2.4. Average precision with document and sentence as domain of cooccurrence and four proximity values. For each collection, the value in boldface is the best value for the collection in this table. Other parameter settings are: df-phrase: 1, df-head: 1, df-comp: 1, length: 2. Percentages are with respect to single term indexing (see Table 2.3).

The effect of domain and proximity. The domain and proximity parameters control the relative location of words that are combined to form a phrase. The effect that varying the domain of cooccurrence has on retrieval effectiveness was tested by experimenting with two values of the domain parameter: document and sentence. The effect of different proximity values was examined by testing a continuum of values between 1 and 30, and in addition allowing unlimited distance between phrase elements.

The figures in Table 2.4 show that variations in proximity have a stronger effect on retrieval performance than different domains have. That is, when proximity is held constant and the domain is varied, only small differences in average precision result. For example, with unlimited proximity, CACM shows a 20.1% increase in average precision using a domain of document, and a 16.2% increase using a domain of sentence. Not surprisingly, the difference is even smaller for more restricted proximities. The largest change in average precision due to different domains of cooccurrence is 6.5% for CISI, when proximity is unlimited. The effect that proximity has on retrieval effectiveness varies from substantial to insignificant. For example, using a domain of document, CACM shows an increase in average precision of 20.1% with unlimited proximity, and an increase of 7.6% with a proximity of 1, for a difference of 12.5%. In contrast, the same parameter settings yield a difference of only 1.4% for MED.

Some general patterns should be noticed with respect to domain of cooccurrence and proximity of phrase elements. Three of the collections, CACM, INSPEC, and CRAN, clearly perform better when the relative location of phrase elements is unrestricted. A domain of document and unlimited proximity is best for these collections. In contrast, CISI performs best with maximally restrictive requirements for the relative location of phrase elements. CISI also differs from the other collections in that the phrases assigned as descriptors under the criteria given in Table 2.4 lead to a reduction, rather than an increase, in average precision, when compared to single term indexing. The MED collection behaves differently from the other collections in that the more restrictive domain of cooccurrence is preferred, while the least restrictive proximity setting is preferred. These differences in average precision for MED are small enough to be considered insignificant, however. A final point is that increases in proximity from 5 upward result, for the most part, in only small changes in average precision. This is an indication that both good and bad phrases are added in approximately equal proportions.

The effect of df-phrase. The df-phrase parameter was used to examine the effect of excluding high and low document frequency phrases from use as phrase descriptors. The experimental results indicate that for most collections, removal of low document frequency phrases has a very small influence on retrieval effectiveness. For example, with $df\text{-}phrase_{min} = 2$, a phrase p is assigned as a descriptor only if $df_p \geq 2$. For this value, and other parameter settings as given in Table 2.4 (domain: document, proximity: unlimited), CISI and MED show very slight increases in average precision of 0.3% and 0.1%, respectively. The only substantial effect was obtained with CACM, where a decrease of 6.1% resulted. For all five collections and this set of parameter values, higher values of df-phrase$_{\text{min}}$ (which result in the exclusion of more low document frequency phrases) yield steadily declining average precision figures. The only possible evidence that exclusion of low document frequency phrases may have a positive effect comes from the MED collection. With other parameter values as given in Table 2.4 (domain: sentence proximity: unlimited), and *df-phrase*$_{min}$ = 3, average precision increases from +3.3% to +4.0%. This increase is too small to be viewed as significant, however.

The results shown in Table 2.2 provide some indication that exclusion of high document frequency phrases can have a positive effect on retrieval performance. The benefit is minimal, however. For example, using 90, 150, 90, and 30 as values of df-phrase$_{max}$ for CACM, INSPEC, CRAN, and CISI, respectively, results in increases in average precision of 0.7% to 4.0% over the best average precision values for these collections shown in Table 2.4. These increases are too small to be regarded as solid evidence that exclusion of high document frequency phrases can lead to substantial improvements in retrieval effectiveness.

A typical example from the CISI collection can be used to illustrate why high document frequency phrases have a negative effect. The phrase descriptor *inform retrief*, usually derived from text phrases like *information retrieval* and *retrieval of information*, contains two elements which themselves have high document frequencies. Because of their high document frequencies and low specificity, single terms such as these tend to have a negative effect on precision. The addition of a phrase descriptor with a relatively high document frequency tends to enhance this negative effect.

Given the observed effect on retrieval performance of excluding both high and low document frequency phrases, it can be concluded that restrictions on the document frequency of phrases cannot be expected to yield significant increases in retrieval effectiveness.

The effect of df-head. The effect of df-head on retrieval performance was examined for each collection by constructing phrases using a large number of different values for this parameter. The maximum value tested for each collection was approximately 10% of the number of documents in the collection. A continuum of smaller values were then tested until a clear pattern could be observed. The largest change in average precision was obtained for CISI, with a value of 50 for df-head and other parameter values as given in Table 2.4 (domain: document, proximity: 1). With a df-head of 1 (that is, with no restrictions on the document frequency of phrase heads), phrase indexing yielded a change in average precision of -2.2% in comparison to single term indexing. With a df-head of 50 this change increased very slightly to -1.5%. Other collections showed either net decreases in average precision, or even smaller positive changes. Since placing restrictions on the document frequency of phrase heads has either an insignificant positive effect, or a negative effect on retrieval performance, it appears that term specificity, as indicated by document frequency, provides little help in identifying terms that should be included in phrases that have been constructed using this approach to phrase indexing.

The effect of df-comp and df-st. Since placing restrictions on the document frequency of phrase heads has little influence on retrieval effectiveness, restrictions on the document frequency of phrase components cannot be expected to have much influence either. This has been verified by a series of experiments that tested various values of df-comp for all of the test collections. The effect of excluding high document frequency single term descriptors was examined by testing a continuum of values for df-st on all collections. For all collections, the effect is very slight; the largest positive effect was for INSPEC, which yielded an increase of 1.4% in average precision.

Some conclusions can be drawn regarding the general applicability of this phrase indexing method:

(1) Under certain circumstances, assignment of phrase descriptors can have a substantial positive effect on retrieval performance. However, the method described here does not consistently yield substantial and statistically significant improvements in retrieval effectiveness for all collections. The range of increase in average precision is from 2.2% to 22.7%. For the collections tested, only CACM and INSPEC show material improvement, while CRAN and MED yield lower levels of improvement that are statistically significant. When applied to CISI, a slight, statistically insignificant increase in performance results.

(2) A single phrase selection strategy is not effective for all collections. This is a serious operational problem, since the most appropriate set of phrase indexing parameter values for an arbitrary collection cannot be determined without extensive experimentation.

CACM, INSPEC, and CRAN perform best when very unrestrictive phrase selection criteria are employed, that is, with the broadest domain of cooccurrence, and unlimited distance between phrase elements. MED can be grouped with CACM, INSPEC, and CRAN, since it performs best with the least restrictive proximity requirement. The difference between MED's performance with a domain of document and sentence is small enough to be disregarded. In contrast, CISI performs best with maximally restrictive phrase selection criteria, where phrase elements must cooccur adjacently in the same sentence. For all collections, further restrictions on the document frequency of phrases and phrase elements (heads and components) have only a slight effect on retrieval performance.

(3) The extreme contrast in the effectiveness of phrase indexing on CACM and CISI can be attributed largely to differences in text characteristics for these collections. In particular, the characteristics of the queries for the two collections differ considerably. CACM queries are primarily short and narrowly focused. CISI queries, however, tend to be considerably longer, more discursive, and not as well focused. Combinations of terms extracted from brief, well-focused statements of information need are more likely to have a positive effect on retrieval performance than combinations of terms extracted from less concise text.

(4) The information about term specificity and relationships among words in text that is provided by document frequency, proximity, and the frequency

of cooccurrence of terms does not provide an adequate basis for a phrase indexing procedure that will consistently yield substantial, statistically significant improvements in retrieval effectiveness. This suggests that a more selective approach to phrase construction is required. The following section presents some evidence that a more selective phrase construction procedure can be developed by making use of more information about relationships among words in text.

2.4. The Quality of Phrase Descriptors

A large sample of phrases generated by the non-syntactic phrase indexing procedure has been examined in order to assess the general quality of the phrases and to analyze the effect they have on retrieval performance. In conducting this analysis, a number of problems with the phrase indexing procedure have become apparent. This section discusses a some of these problems, and outlines possible approaches to solving them.[6] For purposes of illustration, this discussion assumes a restrictive phrase selection strategy like that used for the CISI collection in Table 2.2.

2.4.1. Construction of Inappropriate Phrase Descriptors

A phrase descriptor may be thought of as inappropriate for two general reasons. First, the descriptor may simply not be an accurate indicator of document or query content. Second, the meaning of the source text of a

[6] All examples are taken from experimental document and query collections. The source is given in the text, or in parentheses after each example. For example, (CISI q12) and (CISI

phrase descriptor in a query may differ significantly from the meaning of the source text of a phrase descriptor in a document. This section presents several examples of inappropriate phrase descriptors, explains why they are inappropriate, discusses their effect on retrieval performance, and analyzes the extent to which it may be possible to avoid them or lessen their negative effects.

Phrase indexing consists of two processes: (1) identifying phrases in text, and (2) normalizing the form of phrases that differ in structure, but that are related in meaning. The process of phrase identification has already been explained and illustrated. Normalization is beneficial, since it makes it possible to represent a pair of phrases like *information retrieval* and *retrieval of information* by the single phrase descriptor *inform retrief.* Similarly, the phrases *book review* and *reviews of books* can both be represented by the phrase descriptor *book review.* In non-syntactic phrase indexing, normalization is accomplished by three devices: (1) stemming, (2) regularizing the order of phrase elements, and (3) ignoring stopwords that intervene between content words. All of these devices must be used in order to accomplish the normalization just illustrated.

Although normalization has significant benefits, many of the inappropriate phrase descriptors generated by the non-syntactic phrase indexing process

---

d1340) refer to query 12 and document 1340 in the CISI collection.

are the result of excessive normalization. Several examples are presented below.

Seven queries in the CACM collection contain the text phrase *operating system*, which yields the phrase descriptor *oper system*. In all of these queries, the source of this descriptor is syntactically correct, and the descriptor is a good indicator of document content. A number of documents contain this descriptor, but many of them are related only peripherally, if at all, to the topic of operating systems. The important point illustrated by these examples is that the phrase descriptor *oper system* does not correspond to a single phrase in document and query texts, or even to a set of phrases closely related in meaning:

(1) a fully automatic document retrieval *system operating* on the IBM 7094 is described (CACM d1236)

(2) to illustrate *systems operations* and evaluation procedures (CACM d1236)

(3) extensive data on the *system's operation* (CACM d1533)

(4) to achieve a *system operational* within six months (CACM d2380)

(5) time between project inception and *system operational* date (CACM d1034)

(6) critical to the *system's operating* efficiency (CACM d1226)

(7) examples of overall *system operation* (CACM d3087)

(8) the *system, operated* entirely from a digital display unit, interacts directly with the user (CACM d1695)

(9) the *system* is *operational* and available on the arpa sdc time shared computing system (CACM d1170)

(10) the *system* has been in *operation* (CACM d1665)

(11) the COBOL language was used specifically to enable the *system* to *operate* on three IBM computers (CACM d1168)

(12) the logic required in procedures, *operations*, *systems*, and circuits (CACM d320)

(13) examples of the *operation* of *system* components (CACM d3087)

(14) an *operational system* utilizing this concept (CACM d2919)

(15) the duplex *operation* gives the *system* greater reliability (CACM d252)

The overall effect of this phrase descriptor on retrieval performance for CACM is a reduction in average precision.

Query 25 from the CACM collection is another case in which an appropriate query phrase matches document phrase descriptors constructed from pairs of words that are not related appropriately in the document text. The query contains the phrase in (2.7), which yields the phrase descriptor *comput system*.

(2.7) performance evaluation and modelling of *computer systems*

The source of this descriptor is a syntactically correct noun phrase consisting of *systems* as a head noun and *computer* as a noun phrase premodifier. In document 1591, the phrase in (2.8) also yields the same phrase descriptor.

(2.8) the advantages of this type of *system* for *computer* programming and operation

The source of the phrase descriptor in this case, however is a pair of words that are not related appropriately. Here, *computer* modifies *programming*;

there is no direct syntactic relationship between *computer* and *system*. Further examples illustrating this problem are document 2739, containing the phrase in (2.9) and document 2841, with the phrase in (2.10).

(2.9) a number of *systems* for the *computer* analysis of natural language sentences

(2.10) an experimental *system* for *computer*-aided design

Similarly, document 2325 contains the text in (2.11), which again yields the phrase descriptor *comput system*.

(2.11) these are: foundations (finite precision number *systems*, *computational* complexity), synthesis and analysis of algorithms

There is no syntactic relationship between the phrase elements, and the document is not concerned with the general topic of computer systems.

In all of these examples, the inappropriate document phrase descriptors are the result of a pair of words that happen to occur in close proximity in the text, but that nevertheless are not related syntactically. That is, they do not enter into a relationship of modification with one another. The result is a document phrase descriptor that matches with a query descriptor whose source text differs significantly in meaning from the source of the document descriptor. Rather than being unusual cases, examples of this kind occur frequently in the experimental collections.

Another class of undesirable phrase descriptors results from the construction of phrases from pairs of terms that are related syntactically. The syntactic relationship involved is not appropriate for use as the basis of a phrase descriptor, however, since it is not a relationship of modification. CISI query 24 provides an example of this kind of phrase descriptor. The text phrase in (2.12) contains a pair of conjoined adjectives which together modify the head noun *requirements.*

(2.12) *educational* and *training* requirements

Phrases like *educational requirements* and *training requirements* are syntactically correct and semantically appropriate phrases that could be constructed on the basis of the relationship of modification between the head of the noun phrase and its conjoined modifiers. Each of these phrases refers to a specific kind of requirement. The phrase descriptor *educ train,* however, is constructed from words that do not enter into a relationship of modification with one another and therefore the phrase descriptor does not refer to a more specific concept in the way that *educational requirements* and *training requirements* do. A phrase descriptor of this kind, that is, one derived from a pair of conjoined words, has the effect of giving added weight to a pair of general terms rather than expressing a more precise concept. This conjunction could occur in a wide variety of contexts having to do with education and training, but having little to do with the idea of requirements. Obvious possibilities include *education and training costs* and *education and training programs.* As an actually occurring example, document 692, which is not relevant to query 24, contains the text phrase in (2.13), which yields the phrase descriptor *educ train,* as in query 24.

(2.13) the objective of *education* and *training*

It would be helpful to assign phrase descriptors like *objective of education* and *objective of training* to document 692, since they express more precise concepts. The descriptor derived from *education and training*, however, has the detrimental effect of giving added weight to the quite general descriptors *educ* and *train*. This increases the similarity coefficient for this query-document pair, and raises the rank of the non-relevant document from 15 to 10.

A similar situation is found in CISI query 55, where the text phrase in (2.14), yields the phrase descriptor *anal retrief*, which matches the same descriptor assigned to non-relevant document 454 due to the text phrase in (2.15).

(2.14) the medical literature *analysis* and *retrieval* system

(2.15) information *analysis* and *retrieval*

This match raises non-relevant document 454 from 47 to 29.

A final example indicating the undesirable character of phrase descriptors derived from conjunctions is the descriptor *educ libr*, which is assigned to CISI document 91. The source text is appears in (2.16).

(2.16) the *library* and *educational* community

Even though it has nothing to do with the topic of library education, this phrase could easily match with a query containing the text phrase *library education*, and thus contribute to the increased rank of a non-relevant document.

The inappropriate phrase descriptors discussed in this section can be attributed to five factors: (1) regularization of the order of phrase elements, (2) ignoring intervening stopwords, (3) stemming, (4) construction of phrase descriptors from pairs of words that are not related syntactically, and (5) construction of phrase descriptors from pairs of words that are related syntactically, but that do not enter into a relationship of modification with one another. All of these factors have the potential to result in the construction of phrase descriptors such that a single descriptor may correspond to text phrases that differ greatly in meaning. This in turn will result in inappropriate matches between queries and documents, which will ultimately have a negative effect on retrieval performance.

Problems related to the first three factors could be eliminated simply by abandoning those three normalization techniques. Any benefits resulting from such an inflexible approach, however, would almost certainly be offset by the disadvantages of having no normalization of phrases at all. A better alternative would be to incorporate an approximate phrase matching technique that would take into consideration word order, phrase length, and the morphological structure of phrase elements (Paice and Aragón-Ramírez 1985). Simpler variations on the basic approach could also be attempted. Obvious possibilities include: (1) treatment of conjunctions differently from other non-content words, (2) placing limits on the number of stopwords that may intervene between phrase elements, (3) changing the order of phrase elements only if a stopword intervenes, and (4) taking into consideration punctuation between phrase elements. Even more flexible and selective approachs of this nature, however, would not be able to correctly handle situations in which information about the syntactic structure of text is required.

2.4.2. Failure to Identify Good Phrase Descriptors

The objective of the previous section was to describe and exemplify ways in which the simple criteria of word frequency and proximity lead to the construction of phrases that have a negative effect on retrieval performance. The objective of the current section is to illustrate some common situations in which the non-syntactic phrase indexing process fails to identify phrase descriptors that are good indicators of document or query content and that should have a positive influence on retrieval performance. Whereas simple frequency and proximity criteria often fail to identify useful phrase descriptors, relatively simple syntactic criteria can be used to successfully recognize many appropriate phrases.

Two categories of noun phrases are used for purposes of illustration: (1) noun phrases consisting of adjectival and nominal premodifiers and/or prepositional phrase postmodifiers, and (2) noun phrases involving conjunctions.

From the title in (2.17) non-syntactic phrase indexing would identify the two phrases in (2.18), one correctly, and one incorrectly.[7]

(2.17) the administration of the college library (CISI d14)

(2.18) college library
*college administration

By taking into consideration the syntactic structure of the noun phrase, however, *college library* can still be identified, the incorrect phrase can be avoided, and an additional correct phrase, *library administration*, can by identified. This can be done by simply making use of the fact that *administration* is the head of the noun phrase, that the prepositional phrase *of the college library* modifies *administration*, and that *college* modifies *library* and is not related syntactically to *administration*.

Similarly, for the text phrase in (2.19), non-syntactic phrase indexing could identify the phrases in (2.20). The first of these is correct, but the second is not.

(2.19) the theory of directed graphs (CISI d1385),

(2.20) directed graphs
*theory directed

Again, by using information about the syntactic structure of the phrase, the inappropriate phrase can be avoided, and an additional good phrase, *graph theory* can be generated.

As another example, from the text phrase in (2.21), non-syntactic phrase indexing identifies the phrases in (2.22).

---

[7] An asterisk preceding a phrase indicates that it is considered to be inappropriate.

(2.21) the organization of these library schools (CISI d1423)

(2.22) *library organization
library schools

Here again, one is inappropriate and the other is good. Since the original text phrase has to do with the organization of schools and not the organization of libraries, **library organization* is not appropriate. Knowledge of the syntactic structure of the text phrase makes it possible to avoid identifying the inappropriate phrase, and in addition to identify another correct phrase, *school organization.*

Text phrases of this kind are a potentially rich source of good phrase descriptors. Further, such phrases are not uncommon in the experimental document collections. As an indication of the frequency of such phrases in text, a few additional examples are given in (2.23)-(2.25).

(2.23) the management of large research libraries (CISI d616)

(2.24) targets for research in library education (CISI d1403)

(2.25) evaluation of information retrieval (CISI d829)

The second category of constructions to be considered is noun phrases involving conjunction. Like the complex noun phrases discussed above, these constructions are an important source of good phrase descriptors that cannot adequately be identified on a non-syntactic basis.

Consider, for example, the text phrase in (2.26).

(2.26) parallel and sequential algorithms (CACM q63)

Non-syntactic indexing yields from this the correct phrase *sequential* *algorithms*, and the meaningless **parallel sequential*. Syntactic analysis provides the information that both *parallel* and *sequential* can be understood as modifiers of *algorithms*, thus making it possible to generate two correct phrases, *parallel algorithms* and *sequential algorithms*, and to avoid the inappropriate phrase identified by the non-syntactic procedure.

The same strategy can be applied to more complex constructions. For example, from the text phrase in (2.27), the non-syntactic phrase construction process identifies the phrases in (2.28).

(2.27) the structure, analysis, organization, storage, searching, and retrieval of information (CISI d175),

(2.28)	*structure analysis	*storage searching
	*analysis organization	*searching retrieval
	*organization storage	retrieval information

Five of the six phrases generated are not good indicators of document content. In contrast, syntactic information makes it possible to generate the six good phrases in (2.29), and to avoid constructing all of the inappropriate phrases in (2.28).

(2.29)	information structure	information storage
	information analysis	information searching
	information organization	information retrieval

Like the first category of noun phrases, constructions of this kind are very common in the experimental collections. A few additional examples follow.

(2.30) analysis and perception of shape (CACM q43)

(2.31) the design, operation, and evaluation of retrieval systems (CISI d523)

(2.32) advancement and improvement of the library profession (CISI d22)

(2.33) social, economic, and technological change (CISI d896)

(2.34) working and planned systems for publishing and printing original papers by computer (CISI q7)

The abundance of constructions of this kind is a strong indication that a large number of good phrase descriptors could be identified using syntactic criteria that could not be identified on the basis of frequency and cooccurrence criteria alone, unless a very unrestrictive proximity requirement is used.

CHAPTER 3

SYNTACTIC PHRASE INDEXING

3.1. Introduction

The phrase indexing method examined in chapter 2 takes into consideration only very simple aspects of text structure, namely, term relationships identified by the frequency and cooccurrence characteristics of terms in text. The results of retrieval experiments indicate, however, that that approach to phrase indexing cannot be expected to consistently yield substantial and significant improvements in retrieval effectiveness. From these results it must be concluded that if information about text structure is to be incorporated into a content analysis system in a way that will consistently yield substantial improvements in retrieval effectiveness, then it will be necessary to go beyond simple measures of term frequency and proximity in analyzing text structure.

Several possibilities might be considered in developing further refinements to automatic content analysis methods. These range from semantic analyses based on knowledge about particular domains, to simple approaches to identifying a limited inventory of syntactic patterns (see chapter 4 and references in chapter 1). In the absence of solid experimental evidence, it is difficult to draw conclusions about the potential benefits of using detailed semantic representations of text content for purposes of content analysis in a document retrieval environment. However, there is presently no clear evidence that such detailed analyses can be done accurately enough, and on a scale large enough to be applicable to this environment.

Given the unsolved difficulties related to the more ambitious approaches to text analysis, and the lack of consistent, substantial improvements achieved with the non-syntactic approach to phrase indexing, it is reasonable to investigate an approach to text analysis that is intermediate between the most complex and simplest approaches. This chapter thus has two objectives:

(a) to propose a syntax-based approach to identifying relationships among words in text that can be used to construct phrases for use as content indicators, and

(b) to evaluate this phrase indexing method by performing indexing and retrieval experiments on two experimental document collections.

The objective of syntax-based phrase construction is to use information about the syntactic structure of document and query texts to identify relationships among words that will make it possible to construct useful phrases that could not be correctly identified without syntactic information, and to avoid constructing inappropriate phrases that would be generated with a non-syntactic procedure. Many of the shortcomings of the non-syntactic approach discussed in chapter 2 can be overcome by incorporating syntactic information into the phrase construction process. The approach is intended to be applicable to unrestricted English text, so it does not depend on knowledge of the subject domain of the documents to be analyzed.

The primary text analysis tool is an existing natural language processing system that includes a broad-coverage syntactic grammar, a large general-purpose dictionary, and convenient facilities for manipulating the output of the syntactic analyzer. This phrase indexing method is thus based on:

(a) syntactic structure as determined by the grammar,

(b) information about lexical items as provided by the dictionary, and

(c) specially constructed classes of words that are used in order to be more selective in constructing phrase descriptors.

Use of information about classes of lexical items that are more refined than major grammatical classes (whether based on features from the dictionary, or specially constructed classes), actually goes beyond strictly syntactic information. This kind of information can, in fact, be viewed as a limited kind of semantic information. However, this lexical information is general enough to be applicable to unrestricted text, so using it fits within the requirements of this study.

Certainly much more could be done to refine the content analysis process in general, even without using domain specific information. Some possibilities are mentioned in chapter 5. However, it is necessary to restrict this study to phrase indexing only, in order to place reasonable limits on the scope of the investigation, and to assure that the experimental retrieval results can be fairly compared with the results of the non-syntactic phrase indexing experiments presented in chapter 2.

This chapter is organized as follows. Section 3.2 explains the general approach to syntax-based phrase indexing with simple examples. Section 3.3 provides an overview of PLNLP, the natural language processing system that this implementation of syntactic phrase indexing depends on. Section 3.4 presents complete details of the syntactic phrase indexing method. Section 3.5 is an analysis of the semantic appropriateness of syntax-based phrase descriptors. Section 3.6 summarizes the results of experiments done to test the influence of syntax-based phrase descriptors on retrieval effectiveness.

3.2. Syntactic Phrase Indexing Method—Overview

3.2.1. Decomposition and Normalization

The syntactic phrase construction procedure described here is based on the ideas of decomposition and normalization. The procedure decomposes complex constructions into simpler forms, while preserving much of the information about syntactic relationships among words that is provided by the syntactic analysis system. In addition, the procedure normalizes the form of constructions that differ syntactically, but that are closely related semantically. This is done in such a way that the resulting phrases can be incorporated directly into a vector representation of documents and queries, thus maintaining compatibility with the existing retrieval environment, and avoiding the need to perform complex structure matching operations.

Some simple examples will serve to illustrate the primary characteristics of the approach.[1] To start with, consider the noun phrases in (3.1).

(3.1) automatic text analysis
automatic analysis of scientific text

These two phrases have three words in common. They are closely related in meaning, but their syntactic structures differ significantly, as can be seen from the parse trees in (3.2) and (3.3).

```
(3.2)  NP      AJP     ADJ*    "automatic"
               NP      NOUN*   "text"
               NOUN*   "analysis"
               PUNC    "."

       automatic        analysis
       text             analysis
```

```
(3.3)  NP      AJP     ADJ*    "automatic"
               NOUN*   "analysis"
               PP      PREP    "of"
                       AJP     ADJ*    "scientific"
                       NOUN*   "text"
               PUNC    "."

       automatic        analysis
       text             analysis
       scientific       text
```

In these trees, the head of each constituent is identified by an asterisk. In (3.2), the head of the noun phrase is *analysis*, and there are two premodifiers, an adjective phrase and a noun phrase. The heads of the modifying constructions are the adjective *automatic*, and the noun *text*.

[1] Preliminary versions of this phrase indexing method were discussed in Fagan (1985, 1987).

Phrase construction proceeds simply by combining the head of a construction with the head of each of its modifiers. This yields two simpler phrases, *automatic analysis* and *text analysis*. The original three-word phrase is thus decomposed into two simpler two-word phrases.[2]

In (3.3), the head of the noun phrase is again *analysis*, and the adjective phrase with head *automatic* is again a premodifier. In this case, however, the noun *text* is the head of a prepositional phrase postmodifier, and *text* is itself pre-modified by an adjective phrase with *scientific* as head. Combining the head noun *analysis* with the heads of its premodifier and postmodifier yields the phrases *automatic analysis* and *text analysis*. Within the prepositional phrase, the head *text* is combined with the premodifier *scientific* to yield the phrase *scientific text*.

Although the phrases in (3.2) and (3.3) differ with regard to syntactic structure and lexical content, the simple strategy of associating heads with modifiers accomplishes significant decomposition and normalization. The relatively complex phrases are reduced to simpler, two-word phrases, and two identical phrases are generated from the original constructions. In addition, because the phrase construction procedure takes into consideration the syntactic structure of the phrases, the syntactically and semantically inappropriate phrase *scientific analysis* can be avoided, since its elements are not related

---

[2] As currently implemented, the procedure constructs only two-word phrases. This is not due to any limitations imposed by the syntactic analysis system, or the essential nature of the phrase construction strategy. See section 2.2.1 for further discussion of the motivation for using two-word phrases.

as head and modifier. The non-syntactic approach to phrase construction discussed in chapter 2 would incorrectly generate this phrase.

This method also makes it possible to extract shared components from complex phrases that have similar syntactic structure but contain different lexical items. This is illustrated by the noun phrases in (3.4) and (3.5).

```
(3.4)  NP      AJP     ADJ*     "automatic"
               NP      NOUN*    "document"
               NOUN*   "retrieval"
               PUNC    "."
-------------------------------------------------
       automatic          retrieval
       document           retrieval
-------------------------------------------------
(3.5)  NP      AJP     ADJ*     "automatic"
               NP      NOUN*    "information"
               NOUN*   "retrieval"
               PUNC    "."
-------------------------------------------------
       automatic          retrieval
       information        retrieval
```

The phrase *automatic document retrieval* yields two phrase descriptors, *automatic retrieval*, and *document retrieval*, and the phrase *automatic information retrieval* yields *automatic retrieval*, and *information retrieval.* By decomposing the phrases in this way, the two original phrases yield a common simpler phrase descriptor, *automatic retrieval.* This means that if a query contained the phrase in (3.4), and a document contained the phrase in (3.5), their common sub-phrases, *automatic retrieval,* would match, even though the complete phrases would not. This phrase construction method thus provides a partial-match capability that takes into consideration not only the lexical content of phrases and the order of their elements, but also their syntactic structure. Notice also that the semantically inappropriate descriptors *automatic document* and *automatic information* are avoided because they are not related as head and modifier.

With appropriate extensions and refinements, this simple approach to extracting phrases from syntactic structures can be applied to quite complex constructions, and can successfully identify many useful phrase descriptors and avoid constructing many less desirable ones. The tree structure and phrase descriptors in (3.6), for example, show that good phrase descriptors can be extracted from a complete sentence. Details of the phrase indexing process that make it applicable to more complex constructions are presented in section 3.4. In preparation for this discussion, section 3.3 provides some background about the natural language processing system used to implement this approach to phrase indexing.

```
(3.6) In this paper a probabilistic model of cluster searching based on
      query classification is described.

      DECL    PP      PREP    "in"
                      DET     ADJ*    "this"
                      NOUN*   "paper"
              NP      DET     ADJ*    "a"
                      AJP     ADJ*    "probabilistic"
                      NOUN*   "model"
                      PP      PREP    "of"
                              NP      NOUN*   "cluster"
                              NOUN*   "searching"
                              PTPRTCL VERB*   "based"
                                      PP      PREP    "on"
                                              NP      NOUN*   "query"
                                              NOUN*   "classification"
              VERB    "is"
              VERB*   "described"
              PUNC    "."
```

```
      probabilistic     model
      searching         model
      cluster           searching
      query             classification
```

3.3. PLNLP: A Tool for Natural Language Text Analysis

The phrase indexing method introduced in section 3.2 and the syntactic analysis system it depends on are both implemented in PLNLP, the Programming Language for Natural Language Processing. This section is an overview of the PLNLP system that is intended to briefly explain:

(a) the general features of PLNLP for use as a natural language text analysis tool,

(b) the approach to syntactic analysis used in this system,

(c) why this natural language processing system is well-suited to an application like content analysis for document retrieval, and

(d) how the syntactic phrase indexing strategy is implemented.

Though a detailed treatment of these topics is not required here, a general overview will make later discussion of the phrase indexing method more easily understandable. Complete details of the system can be found in Heidorn (1972), and a shorter discussion is available in Heidorn (1975). This section draws on both of these sources. Langendoen and Barnett (1986) is a tutorial introduction to PLNLP from a linguist's perspective.

As a programming language, PLNLP is a language for writing augmented phrase structure rules in which the entities specified on the left-hand side of a rule are re-written as the entities specified on the right-hand side of the rule.[3] Each rule can be augmented by specifications that state the conditions under which the rule can be applied, and the structure-building actions that are to be taken when the rule is applied. In the implementation used for this study, PLNLP rules are translated into LISP/VM. The scope of what can be accomplished with PLNLP rules is essentially unlimited, since the language provides constructs that make it possible to do with PLNLP rules anything that can be done with LISP.

PLNLP augmented phrase structure rules perform two functions: decoding and encoding. Decoding is the process of converting natural language text into a structured format that explicitly represents relationships among text elements. This structured format is represented by a data structure

[3] See Winograd (1983:377-383) for a general discussion of the augmented phrase structure formalism.

called a record. Records are sets of attribute-value pairs that can represent entities of varying complexity, for example, letters, words, nouns, verb phrases, and sentences. Since the value of an attribute of one record can be a pointer to another record, entities involving complex relationships can be represented. Encoding is the process of converting these structured representations into some form of text.

Decoding thus corresponds to natural language analysis, or parsing, whereas encoding corresponds to synthesis, or generation. Decoding rules are augmented phrase structure rules that specify how text is to be converted into record structures. Encoding rules are augmented phrase structure rules that specify how record structures are to be converted into text.

As a natural language processing system, PLNLP provides facilities for applying decoding rules for text analysis, and applying encoding rules for text synthesis. The decoding (parsing) algorithm uses a bottom-up, left-to-right strategy that produces all parses of a text string in parallel. The encoding algorithm uses a top-down, serial approach.

3.3.1. Syntactic Parsing with PLNLP

Jensen (1986) provides a thorough discussion of the general approach taken in writing the PLNLP English Grammar (PEG). The brief overview presented here is based primarily on this source. Other discussions of the grammar and its applications can be found in Heidorn et al. (1982), Richardson (1985), Jensen (1987), Miller, Heidorn, and Jensen (1981), and Miller (1980).

In developing the PLNLP grammar, one objective has been to provide the capability of dealing in a useful way with unrestricted English text. Two characteristics of the grammar are essential to approaching this objective. First, extensive semantic information is not used, since it is not realistic to assume that such information will be available for all subject areas that might be encountered when dealing with unrestricted text. Thus the grammar presently uses only information about the syntactic characteristics of words.[4] Second, the system must have few limitations regarding the vocabulary it is capable of handling. This requirement is met by an online dictionary containing word-class and other syntactic information for about 130,000 entries.[5]

In order to be robust enough to deal adequately with unrestricted text, a natural language processing system must be able to handle three aspects of parsing. Jensen (Jensen 1986:6-7; Jensen and Heidorn 1983:93; Jensen et al. 1983:147-148) refers to these as (a) core grammar, (b) parsing ambiguity, and (c) parsing failure. The core grammar is a set of 235 decoding rules that define the primary, generally well-understood grammatical structures of

[4] In principle, there is nothing to prevent semantic information from being used in the parsing process. In fact, semantic constraints can be readily incorporated into PLNLP rules, if such information is available (Jensen and Binot 1987).

[5] For the present study, a subset of the complete dictionary was used. This reduced dictionary contains only entries for words that occur in the experimental document and query collections.

English. Parsing ambiguity arises when the core grammar yields more than one parse for an input string. This is dealt with by a peripheral procedure that ranks the alternative parses by calculating a parse metric for each parse (Heidorn 1982). The alternative parses are then presented in rank order, starting with the most highly valued parse. Parsing failure arises when an input string does not correspond to a structure defined by the core grammar. This is handled by a strategy of *parse fitting*, which examines the records constructed when the parse was attempted, and selects records that are likely to provide the most useful information about the syntactic structure of the input string. These records are then available as a substitute for a successful parse.

3.3.2. Document Content Analysis Using PLNLP and PEG

The PLNLP system is well-suited to an application like content analysis for document retrieval for three primary reasons. First, the PLNLP grammar covers a broad range of English constructions, and is not dependent on semantic information about a particular subject domain. Second, the system deals with multiple parses and failed parses in a useful way. When multiple parses are produced, the parse metric provides a way to select one parse that is likely to be useful for further processing. When parsing is unsuccessful, the fitting procedure typically identifies some lower level structures (such as noun phrases and prepositional phrases) from which useful information can be gathered for document indexing. Finally, encoding rules are a convenient mechanism for systematically examining the record structures that represent a parsed sentence and selectively extracting elements that may be useful for representing document content.

3.3.3. Using PLNLP Encoding Rules for Phrase Indexing

The approach to phrase indexing introduced in section 3.2 has been implemented with PLNLP encoding rules. The purpose of this section is to explain, in a greatly simplified way, the essence of how this is done. Encoding rules used for phrase indexing are called *phrase indexing rules*, or *phrase construction rules.*

Though complete details of the implementation cannot be included here, it is hoped that this overview indicates that encoding rules provide a convenient and powerful means of manipulating the complex record structures that are used to represent syntactically analyzed natural language text. Because encoding rules provide this capability, they are uniquely well-suited to the task of document content analysis in general.

Parsing is accomplished by having the decoding algorithm apply decoding (grammar) rules to input text. The primary result of the decoding (parsing) process is a structure consisting of a set of records connected by pointers from one record to another. This record structure explicitly represents the relationships among the elements of the input string, as determined by the decoding rules.

The general structure of the relationships specified by a record structure can be represented by a tree diagram. Since most of the structural characteristics of syntactically analyzed text that are essential to the phrase indexing rules correspond closely to the structural relationships represented by tree diagrams, it is sufficient to discuss the implementation of phrase indexing rules in terms of their operation on trees, rather than their operation on the more complex underlying record structures. Thus instead of talking about phrase indexing rules applying to a record of a certain kind, it is sufficiently accurate to talk about a rule applying to a particular kind of node in a phrase structure tree.

Figure 3.1 illustrates the basic form of encoding rules, which is essentially the same as that of production rules. Each rule consists of a rule number, a left-hand side, an arrow, and a right-hand side. The left-hand side of a rule identifies the type of object that the rule can be applied to. The right-hand side of the rule specifies the objects that the rule is to create. Rule (0) in part (a) of Figure 3.1 is a simple unaugmented encoding rule that applies to any NP, and produces a DET and a NOUN. Rule (0') in part (b) of Figure 3.1 is an augmented form of Rule (0). The left-hand side of this rule is augmented with condition specifications. The condition specifications state further requirements that must be fulfilled by a particular NP in order for the rule to apply to it. The elements on the right-hand side of Rule (0') are augmented with action specifications. Action specifications generally give further instructions specifying how the named objects (DET and NOUN) are to be created.

The simplified rules in Figure 3.2, together with the trees and phrase descriptors in Figure 3.3, illustrate how rules of this form can be used to implement the phrase indexing method introduced in section 3.2.[6] But before explaining these phrase indexing rules, it is necessary to introduce some terminology that is needed for talking about tree diagrams and how rules apply to them.

---

(0) NP → DET NOUN

(a) A simple, unaugmented encoding rule.

(0') NP(Condition1, Condition2) →

DET(Action1)

NOUN(Action1, Action2)

(b) An encoding rule augmented with condition and action specifications.

FIGURE 3.1. Basic form of encoding rules.

[6] The rules in Figure 3.2 do not conform exactly to the syntax of PLNLP rules. These rules are greatly simplified, and are intended only to provide a general indication of how encoding rules can be used to selectively extract elements from tree structures and use these elements to construct phrase descriptors.

(1)	NP(Premodifiers)	→	
		OUTPUT(Top[Premodifiers], Head)	
		NP(Premodifiers=Rest[Premodifiers])	
(2)	NP	→	NULL

FIGURE 3.2. Simplified encoding rules (phrase indexing rules) for constructing phrase descriptors from noun phrases.

Parse Tree				Phrase Descriptors (output)
NP	AJP	ADJ*	automatic	—
	NP	NOUN*	text	
	NOUN*	analysis		
(a) The original noun phrase.				
NP	NP	NOUN*	text	automatic analysis
	NOUN*	analysis		
(b) After first application of Rule (1).				
NP	NOUN*	analysis		text analysis
(c) After second application of Rule (1).				

FIGURE 3.3. Decomposition of the noun phrase *automatic text analysis* and construction of phrase descriptors using the encoding rules in Figure 3.2.

Each constituent in a tree that represents a construction other than a lexical category has a *head*.[7] In the tree diagrams, the head of a construction is identified by an asterisk. For example, the leftmost NP in the tree diagram in part (a) of Figure 3.3 represents the noun phrase *automatic text analysis*. Its head is the NOUN *analysis*. The head of a construction may have *premodifiers* and/or *postmodifiers*. The NOUN *analysis* in this figure has two premodifiers, an AJP and a NP, but it has no postmodifiers. The top (or first) premodifier is the one that would appear in the left-most position in a standard textual representation of the construction; this is the constituent that would be spoken first. The bottom (or last) premodifier occurs in the right-most position; this constituent would be spoken last. Thus in Figure 3.3 (a), *automatic* is the top (first) premodifier of the head *analysis*, and *text* is the bottom (last) premodifier of the head *analysis*. The same terminology applies to postmodifiers.

Before the rules in Figure 3.2 begin applying to the noun phrase *automatic text analysis* in Figure 3.3, the noun phrase has the structure shown in part (a) of that figure. At this point, no phrase descriptors have been constructed, so the column labeled "Phrase Descriptors" is empty. As indicated by their left-hand sides, both rules in Figure 3.2 apply to noun phrases, that is, constructions represented as NP nodes in parse trees. The

[7] Examples of lexical categories are NOUN, PREP, and VERB, each of which usually corresponds to a single word. Non-lexical categories, such as NP, VP, and PP, represent higher-level constituents.

rules differ, however, in that the left-hand side of Rule (1) is augmented with a condition specification, whereas the the left-hand side of Rule (2) has no condition specifications. The encoding algorithm attempts to apply rules in the order given, so an attempt would be made to apply Rule (1) before Rule (2). The first rule whose condition specifications are fulfilled is applied.

The condition "Premodifiers" in Rule (1) stipulates that this rule can be applied to an NP only if it has at least one premodifier. The NP in (a) does have premodifiers, so the rule applies. When the rule applies, two things happen: (1) some output is produced, and (2) a new NP node is constructed. These actions are indicated by 'OUTPUT' and 'NP' on the right-hand side of the Rule (1). The augmentations associated with OUTPUT specify the kind of output to be produced. The specification "Top[Premodifiers]" is a procedure call that returns the top (first) premodifier in this noun phrase, so the call returns *automatic*. The specification "Head" simply refers to the head of this noun phrase, *analysis*. These two elements are then written out as the phrase descriptor *automatic analysis*. The augmentation associated with the NP on the right-hand side of Rule (1) specifies how the new NP node is to be constructed. By default, an NP on the right-hand side of a rule starts out as an exact copy of the NP on the left-hand side of that rule. The specification

Premodifiers=Rest[Premodifiers]

then alters the structure of the NP by removing its top (first) premodifier. The procedure call "Rest[Premodifiers]" returns the original list of premodifiers with the first element removed.

The end result of this application of Rule (1) is that a new NP *text analysis* has been constructed, having the structure shown in part (b) of Figure 3.3. In addition, the phrase descriptor *automatic analysis* has been produced, as shown in the phrase descriptor column of that figure.

Since the NP in part (b) has a premodifier, *text*, Rule (1) applies to it as it applied to the NP in (a). The result is a new NP *analysis* and a phrase descriptor *text analysis*, as shown in part (c) of Figure 3.3. This new NP has no premodifiers, so it does not fulfill the condition specifications of Rule (1). Since it has no condition specifications, Rule (2) does apply. A right-hand side of "NULL", as in Rule (2), produces no output, and signals the end of processing for this construction.

The PLNLP programming environment includes facilities that make it easy to traverse the record structure that represents a parse tree, thus making it a simple matter to apply phrase indexing rules to potentially every node in a tree. The condition specifications on encoding rules provide the added capability of extracting phrases selectively, using only certain kinds of constructions as sources of phrase descriptors, if desired.

The approach to decomposing parse trees illustrated above is the basis for the phrase indexing method introduced in section 3.2 and discussed further in section 3.4. The refinements to be presented in section 3.4 have been implemented with encoding rules of the same basic form as those in Figure 3.2.

Though these refinements introduce substantial complexity into the rules, the purpose of the refinements is conceptually quite simple. The purpose is to be more selective in extracting elements from parse trees for use a phrase descriptors.

3.4. Syntactic Phrase Indexing Method—Details

Section 3.2 explained in general terms the objectives of syntax-based phrase indexing, and introduced the essential elements of the the phrase construction strategy. Section 3.3 provided an introduction to how encoding rules can be used to manipulate parse trees and extract elements from them that are useful for purposes of phrase indexing. This section describes the phrase indexing process in greater detail. Included are discussions of

(a) the major grammatical constructions treated by the phrase indexing rules,

(b) how phrase descriptors are extracted from these constructions, and

(c) some refinements to the basic strategy of associating construction heads with modifiers that allow additional useful phrase descriptors to be identified.

Though the refinements to the phrase indexing method presented in this section have been implemented with PLNLP encoding rules, an effort has been made to describe the process using terminology that is largely independent of the details of the PLNLP implementation. Thus the discussion assumes only the following:

(a) the existence of a rule-based language allowing for flexible statement of both conditions on rule application and actions for structure building,

(b) some familiarity with phrase structure trees,

(c) the definitions of *head*, *premodifier*, and *postmodifier*, as given in section 3.3.3, and

(d) some familiarity with general linguistic concepts.

3.4.1. Selection of Construction Types

The general approach to phrase indexing used in this study is based on the ideas of decomposition and normalization. In order to construct phrase descriptors of good quality, however, decomposition must be done selectively. This is done by restricting the application of phrase indexing rules so that only certain kinds of constructions are allowed to participate in the phrase construction process.

The most complex syntactic construction dealt with here is the sentence, so the first step in decomposition is to select from each sentence those elements that are likely to be good sources of phrase descriptors. This is done by selecting a few major syntactic construction types that will be analyzed further by rules developed for each type of construction. This initial selective step can be viewed as a coarse-grained filter that eliminates a significant proportion of the constituents of a sentence at relatively little cost.

The constructions used as sources of phrase descriptors are selected subclasses of: (1) noun phrases, (2) prepositional phrases, (3) adjectival constructions, and (4) verbal constructions. Corresponding to each of these construction types is a set of encoding rules that determine how phrase descriptors are to be constructed from each type.

3.4.2. Noun Phrases

For purposes of document content analysis, noun phrases are generally considered to be a good source of words and phrases that are valuable indicators of document content. Some evidence has been gathered to support this point of view (Waldstein 1981). For this reason, this study has been directed primarily toward constructing phrase descriptors based on noun phrases. The following subsections describe methods of further restricting the set of nominal constructions used as sources of phrase descriptors, and the methods developed to extract as many good phrase descriptors as possible from the selected constructions.

3.4.2.1. Restrictions on Heads and Modifiers

The basis of the syntactic phrase indexing strategy is the idea of constructing phrase descriptors that consist of the head of a construction together with the head of a modifier. This approach does yield a number of good phrases. But if applied without further restriction, it also yields a large number of phrases that are not good indicators of document content, and therefore are not useful as phrase descriptors. Many undesirable phrases can be avoided by using condition specifications in indexing rules that prevent the rules from applying to constructions having certain characteristics. Such restrictions can be placed on both heads and modifiers.

Restrictions of this kind should be stated so that they apply to general categories whenever possible, and should be applicable to all kinds of text. In order for the phrase indexing rules to be generally applicable, rules must not use restrictions that are suitable only for text dealing with a specific subject area. Three general kinds of information have been used to specify that a head or modifier in a given construction should not be included as an element of a phrase descriptor. These include:

(a) syntactic category features, as determined by the syntactic analysis system,

(b) lexical features, as specified by the dictionary entry for a given word, and

(c) membership of words in various classes of lexical items that have been defined specifically for the purpose of constructing phrases for use as content indicators.

The noun phrase in (3.7) illustrates a typical situation in which syntactic categories can be used to avoid constructing undesirable phrase descriptors.

```
(3.7)  NP      DET     ADJ*     "the"
               NOUN*   "efficiency"
               PP      PREP     "of"
                       DET      ADJ*     "these"
                       QUANT    ADJ*     "four"
                       AJP      ADJ*     "sorting"
                       NOUN*    "algorithms"
               PUNC    "."
```

Without further restrictions, a noun phrase like the one in (3.7) would yield the phrases in (3.8).

(3.8)	**Modifier**	**Head**
	the	efficiency
	algorithms	efficiency
	these	algorithms
	four	algorithms
	sorting	algorithms

Of these five phrases, only two could be considered good content indicators. The phrase descriptors with modifiers *the*, *these*, and *four* should be excluded. This can be accomplished by placing conditions on the application of rules which prevent the syntactic categories DET (determiner) and QUANT (quantifier) from being included as elements of phrase descriptors. Since a condition stated in terms of syntactic categories is very general, a single condition can exclude a large number of inappropriate phrase descriptors. A few additional examples are: *all levels*, *some extent*, and *many journals*.

Associated with many lexical items in the dictionary are lexical features. The feature NUM, for example, is associated with words that represent cardinal numbers. Just as conditions on phrase construction can refer to syntactic categories, lexical features can also be used. A condition stating that a noun phrase whose head has the lexical feature NUM cannot be used as the source of a phrase descriptor, prevents a useless phrase descriptor like *causes one* from being generated from the noun phrase *one of the main causes*. Similarly, the feature ORD is associated with words that represent ordinal numbers. This feature can be used to avoid undesirable phrase descriptors like *third procedure*, which would otherwise be constructed from the text phrase *the third clustering procedure*.

It is preferable to specify restrictions of this kind by using syntactic categories or lexical features, so that the restrictions will be as general as possible. Nevertheless, it is still necessary to explicitly exclude individual lexical items when syntactic categories and lexical features are not sufficient to define the desired set. For example, there are a number of semantically empty nouns that should not be used in phrase descriptors. These include, *kind, less, more, most, other, same, use, using,* and *way*. By excluding these words from use as heads of phrase descriptors, undesirable phrases like *schemes other* and *power kinds*, which would otherwise be constructed from the text phrases in (3.9) and (3.10), can be avoided.

(3.9) related to each other in meaningful schemes

(3.10) two kinds of power

3.4.2.2. Treatment of Conjoined Modifiers

The basic strategy of constructing phrases by associating the head of a constituent with the head of each of its modifiers is adequate when each modifier is a simple constituent. In order to to be fully general, however, it is necessary to deal with more complex constructions, for example, constructions in which a modifier consists of two or more conjoined constituents.

The noun phrase in (3.11), *library and information networks*, consists of a head noun *networks* and a single noun phrase premodifier *library and information*. As indicated in the parse tree, the conjunction *and* is analyzed by the PLNLP grammar as the head of this noun phrase premodifier. Since *and* is the head of the premodifier, in their simplest form, the phrase construction rules would yield the phrase *and networks*. To avoid constructing this phrase, it is necessary to treat modifiers that have conjunctions as heads differently from other modifiers.

An effective approach is to associate the head of each conjunct of the modifying constituent with the head of the noun phrase. This is done by altering the structure of the noun phrase in (3.11) so that the conjuncts, *library* and *information*, both become premodifiers of *networks*, rather than modifiers of the conjunction *and*. The result is a noun phrase with the structure shown in (3.12). After doing this, the two good phrases *library networks* and *information networks* can be constructed in the usual way. Conjoined postmodifiers are treated similarly.

```
(3.11)  NP      NP      NP      NOUN*   "library"
                        CONJ*   "and"
                        NP      NOUN*   "information"
                NOUN*   "networks"
                PUNC    "."
```

```
(3.12)  NP      NP      NOUN*   "library"
                NP      NOUN*   "information"
                NOUN*   "networks"
                PUNC    "."
```

This generalization of the phrase construction strategy is important, since constructions of this type are a source of good phrase descriptors, and such constructions occur commonly in document and query texts. Further examples appear in (3.13), (3.14), and (3.15).

```
(3.13)  NP      AJP     AJP     ADJ*    "physical"
                        CONJ*   "and"
                        AJP     ADJ*    "medical"
                NOUN*   "sciences"
                PUNC    "."
-----------------------------------------------------
        physical          sciences
        medical           sciences
```

```
(3.14)  NP      AJP     AJP     ADJ*    "scientific"
                        CONJ*   "and"
                        AJP     ADJ*    "technical"
                NOUN*   "publication"
                PUNC    "."
-----------------------------------------------------
        scientific        publication
        technical         publication
```

```
(3.15)  NP      NP      NOUN*   "information"
                NOUN*   "dissemination"
                PP      PREP    "by"
                        NP      NOUN*   "journals"
                        CONJ*   "and"
                        NP      NOUN*   "periodicals"
                PUNC    "."
-----------------------------------------------------
        information       dissemination
        journals          dissemination
        periodicals       dissemination
```

3.4.2.3. Treatment of Conjoined Noun Phrases

Just as the phrase construction rules must be general enough to handle conjoined modifiers, they must also deal in a useful way with conjoined noun phrases. The noun phrase in (3.16), for example, consists of three conjoined noun phrases which have *status*, *position*, and *function* as heads.

```
(3.16)  NP      DET     ADJ*    "the"
                NP      NOUN*   "status"
                CONJ    ","
                NP      NOUN*   "position"
                CONJ*   ", and"
                NP      NOUN*   "function"
                        PP      PREP    "of"
                                DET     ADJ*    "the"
                                NOUN*   "librarians"
                PUNC    "."
```

```
        librarians      status
        librarians      position
        librarians      function
```

The last of these has a prepositional phrase *of the librarians* as postmodifier. Without further elaboration, the phrase construction rules would treat the head of this noun phrase, "*, and,*" just as any other head, yielding the four phrase descriptors in (3.17).

(3.17)	**Modifier**	**Head**
	status	, and
	position	, and
	function	, and
	librarians	function

The three descriptors containing the conjunction as head are of no use as content indicators, and must be avoided.

Since the meaning of this noun phrase has to do with the status of librarians and the position of librarians in addition to the function of librarians, two additional phrase descriptors shown in (3.18) should also be generated.

(3.18)	**Modifier**	**Head**
	librarians	status
	librarians	position

This can be accomplished simply by associating the modifier of the last conjunct, *librarians*, with the head of each of the three conjuncts of the noun phrase.

Distributing the modifier of one conjunct of a noun phrase over all conjuncts in this way yields a large number of good phrase descriptors. This is illustrated further by the examples in (3.19) and (3.20). Rather than the single phrase *workers skills* being constructed from *the attitudes and skills of traditional workers* in (3.19), this approach also yields *workers attitudes*. Similarly in (3.20), two phrases, *interface philosophy* and *interface design*, are produced in addition to *interface implementation*.

```
(3.19)  NP      DET     ADJ*    "the"
                NP      NOUN*   "attitudes"
                CONJ*   "and"
                NP      NOUN*   "skills"
                        PP      PREP    "of"
                                AJP     ADJ*    "traditional"
                                NOUN*   "workers"
                PUNC    "."
```

```
        workers             attitudes
        workers             skills
        traditional         workers
```

```
(3.20)  NP      DET     ADJ*    "the"
                NP      NOUN*   "philosophy"
                CONJ    ","
                NP      NOUN*   "design"
                CONJ*   ",      and"
                NP      NOUN*   "implementation"
                        PP      PREP    "of"
                                DET     ADJ*    "an"
                                AJP     ADJ*    "experimental"
                                NOUN*   "interface"
                PUNC    "."
```

```
        interface           philosophy
        interface           design
        interface           implementation
        experimental        interface
```

In order to avoid constructing inappropriate phrases, however, this strategy can be applied only under restricted circumstances. The need for further restrictions is illustrated by the noun phrase in (3.21). The meaning of this noun phrase is such that it is not correct to distribute the modifier of each of the conjuncts over both conjuncts. That is, the phrases *program performance* and *human consistency* should be constructed, but the phrases *program consistency* and *human performance* should not be constructed.

```
(3.21)  NP      NP      NP      NOUN*   "program"
                        NOUN*   "performance"
                CONJ*   "and"
                NP      AJP     ADJ*    "human"
                        NOUN*   "consistency"
                PUNC    "."
```

```
        program             performance
        human               consistency
```

In a large proportion of the cases examined, the following restriction yields appropriate phrases:

> Only the premodifiers of the first conjunct, and the postmodifiers of the last conjunct can be distributed. In addition, the premodifier of the first conjunct is distributed only if none of the other conjuncts has a premodifier. Similarly, the postmodifier of the last conjunct is distributed only if none of the other conjuncts has a postmodifier.

In a case like (3.22), these criteria allow both a premodifier and postmodifier to be distributed over all conjuncts, yielding six good phrase descriptors rather than two.

Without discussing all the details of how this is done, the essence of the process is that from a noun phrase like the one in (3.22), the three simpler noun phrases shown in (3.23), (3.24), and (3.25) are constructed. The phrase indexing rules then apply to these simpler noun phrases in the usual way, yielding the expected phrase descriptors.[8]

[8] Note that construction of the simpler noun phrases in (3.23), (3.24), and (3.25) is done mainly with pointers to existing structures, so extensive copying of tree structures is not required.

```
(3.22)  NP      NP      AJP     ADJ*    "automatic"
                        NOUN*   "analysis"
                CONJ    ","
                NP      NOUN*   "storage"
                CONJ*   ",      and"
                NP      NOUN*   "retrieval"
                        PP      PREP    "of"
                                NOUN*   "information"
                PUNC    "."
```

```
        automatic                 analysis
        information               analysis
        automatic                 storage
        information               storage
        automatic                 retrieval
        information               retrieval
```

```
(3.23)  NP      AJP     ADJ*    "automatic"
                NOUN*   "analysis"
                PP      PREP    "of"
                        NOUN*   "information"
                PUNC    "."
```

```
        automatic                 analysis
        information               analysis
```

```
(3.24)  NP      AJP     ADJ*    "automatic"
                NOUN*   "storage"
                PP      PREP    "of"
                        NOUN*   "information"
                PUNC    "."
```

```
        automatic                 storage
        information               storage
```

```
(3.25)  NP      AJP     ADJ*    "automatic"
                NOUN*   "retrieval"
                PP      PREP    "of"
                        NOUN*   "information"
                PUNC    "."
```

```
        automatic                 retrieval
        information               retrieval
```

3.4.3. Prepositional Phrases

Given the approach to constructing phrase descriptors from noun phrases presented in the previous section, prepositional phrases can be handled directly and simply. The noun phrase object of a preposition is extracted from the prepositional phrase, and the resulting noun phrase is then processed by the phrase indexing rules exactly as any other noun phrase would be.

For example, from the prepositional phrase in (3.26), the noun phrase in (3.27) is constructed, yielding two phrase descriptors, *information exchange* and *information dissemination.*

```
(3.26)  PP      PREP    "for"
                NP      NOUN*   "exchange"
                CONJ*   "and"
                NP      NOUN*   "dissemination"
                                PP      PREP    "of"
                                        NOUN*   "information"

(3.27)  NP      NP      NOUN*   "exchange"
                CONJ*   "and"
                NP      NOUN*   "dissemination"
                                PP      PREP    "of"
                                        NOUN*   "information"

        information       exchange
        information       dissemination
```

3.4.4. Adjectival Constructions

A significant proportion of adjectival constructions are not likely to be important indicators of document content in isolation from the nouns they modify. However, many adjective phrases that consist of an adjective with an adverbial modifier do provide a source of useful phrase descriptors. The noun phrase in (3.28), for example, contains two adjective phrases of this kind. The phrase descriptors *automatically matching* and *automatically drawing* can be extracted from these phrases.[9]

```
(3.28)
NP      DET     ADJ*    "a"
        NOUN*   "system"
        PP      PREP    "for"
                AJP     AJP     ADJ*    "encoding"
                        CONJ    ","
                        AJP     AVP     ADV*    "automatically"
                                ADJ*    "matching"
                        CONJ*   "and"
                        AJP     AVP     ADV*    "automatically"
                                ADJ*    "drawing"
                AJP     ADJ*    "chemical"
                NOUN*   "structures"
```

```
        automatically     matching
        automatically     drawing
```

Though the parse in (3.28) does yield these two good phrase descriptors, it is clear from the meaning of this sentence that this analysis is not the most appropriate one. A more appropriate parse would have *matching* and *drawing* analyzed as verbs, as in (3.29). Since this kind of ambiguity cannot be resolved on syntactic grounds, the grammar produces both of these analyses. It is important for the phrase indexing rules to correctly handle both, however, since the parse metric assigns a higher rank to the parse in (3.28) than the parse in (3.29). If phrase descriptors are taken only from the highest

[9] **Other phrase descriptors are also constructed, but only those directly relevant to the current discussion have been included in (3.28), (3.29), (3.30), and (3.31).**

ranking parse, it is important to have the phrases generated from the structure in (3.28), even though it is not the best analysis.

```
(3.29)  NP      DET     ADJ*    "a"
                NOUN*   "system"
                PP      PREP    "for"
                                VP      VERB*   "encoding"
                                CONJ    ","
                                AVP     ADV*    "automatically"
                                VP      VERB*   "matching"
                                CONJ*   "and"
                                VP      AVP     ADV*    "automatically"
                                        VERB*   "drawing"
                                        NP      AJP     ADJ*    "chemical"
                                                NOUN*   "structures"
```

```
        automatically     matching
        automatically     drawing
```

Constructing phrase descriptors from adjectival constructions of this kind is also valuable, because expressions that are analyzed by the grammar in this way have common nominal and verbal paraphrases, and it is important for the phrase construction rules to be able to normalize all of these semantically related constructions to the same form. For example, the noun phrase in (3.30) is very similar in meaning to the one in (3.28), but in (3.30), *matching* and *drawing* are nouns rather than adjectives, and are modified by the adjective *automatic* rather than an adverb. After a stemming operation is performed to regularize morphological variants, however, the phrase descriptors generated from these two noun phrases will be identical.

```
(3.30)  NP      DET     ADJ*    "a"
                NOUN*   "system"
                PP      PREP    "for"
                                AJP     ADJ*    "automatic"
                                AJP     ADJ*    "chemical"
                                NP      NOUN*   "structure"
                                NP      NOUN*   "matching"
                                CONJ*   "and"
                                NP      NOUN*   "drawing"
```

```
        automatic         matching
        automatic         drawing
```

A comparable situation arises in a sentence such as (3.31). Here, an equivalent idea is expressed in an infinitival construction from which the phrase descriptors *automatically match* and *automatically draw* are derived.

```
(3.31)  DECL    NP      DET     ADJ*    "the"
                        NOUN*   "system"
                VERB    "is"
                VERB*   "designed"
                INFCL   INFTO   "to"
                        AVP     ADV*    "automatically"
                        VP      VERB*   "match"
                        CONJ*   "and"
                        VP      VERB*   "draw"
                                NP      AJP     ADJ*    "chemical"
                                        NOUN*   "structures"
                PUNC    "."
```

```
        automatically     match
        automatically     draw
```

Unlike the noun phrase in which these expressions have both an adjectival analysis and a verbal analysis ((3.28) and (3.29)), the constructions in (3.30) and (3.31) each have only a single analysis. It would not be possible to normalize the form of all of these semantically related expressions if the phrase indexing rules were not capable of dealing with adjectival constructions having adverbial premodifiers.

Though adjectival constructions like those in (3.28) are useful, many other adjective phrases are clearly not good sources of phrase descriptors. For example, phrases like *most suited*, *completely unnecessary*, and *ultimately important*, are not important content indicators, and are not likely to have a significant positive effect on retrieval effectiveness. A simple restriction suffices to exclude most undesirable phrases, and to include many useful ones. First, it is required that the head of the adjective phrase be a present or past participle. This allows phrases like those in (3.28), as well as phrases like *alphabetically arranged*. A small class of participles is then explicitly excluded, for example, *interesting*. In addition, several classes of adverbial modifiers are excluded. These include comparatives (e.g., *less*, *more*), superlatives (e.g., *most*), intensifiers (e.g., *very*), and specifiers (e.g., *mainly*, *mostly*). These can be identified by the lexical features provided by the dictionary. In addition to modifiers excluded on this basis, a large class of adverbs judged to be of little importance as content indicators are also excluded. The base forms of these adverbs appear in Figure 3.4.[10]

[10] This list was compiled by examining the vocabulary of the CISI document collection.

also	concomitant	full	name	short
ably	consequent	general	near	similar
absolute	considerable	genuine	new	special
abundant	converse	good	nice	sure
accidental	current	great	normal	thorough
accordingly	deep	gross	obvious	time
actual	definite	habitual	occasional	tremendous
admirable	different	hard	open	true
admittedly	dramatic	high	over	undeniable
alternate	especial	hopeful	overwhelming	undoubted
alternative	essential	ideal	particular	unenviable
apparent	eventual	immediate	possible	unerring
appropriate	evident	important	preferable	unexpected
bare	exceeding	incidental	present	unforgivable
basic	exclusive	inevitable	presumably	unfortunate
blind	extensive	invariable	probable	unlike
brief	extreme	keen	pure	unlikely
broad	favorable	kind	rare	utter
careful	favourable	large	ready	various
cautious	final	late	real	wanton
certain	fine	light	remarkable	wholehearted
cheerful	firm	like	remarkably	whole
clear	first	main	respective	wholly
common	former	marked	rich	wide
complete	fortunate	mere	right	
concise	frank	most	seeming	

FIGURE 3.4. Adverbial base forms excluded from use as modifiers in phrase descriptors.

3.4.5. Verbal Constructions

The syntax-based phrase indexing method proposed in this study is directed primarily toward extracting phrase descriptors of good quality from nominal constructions. However, since nominal constructions commonly include various kinds of verbal constituents as modifiers, generating phrase descriptors from noun phrases cannot be done adequately without dealing with verbal constructions, at least to a limited extent. For this reason, indexing rules have been included to deal with a limited inventory of verbal constructions, namely, infinitival and participial clauses occurring as postmodifiers of nouns. A comprehensive treatment of verbal constructions has not yet been implemented.

3.4.5.1. Clauses as Postmodifiers of Nouns

Noun phrases with present participial, past participial, and infinitival clauses as postmodifiers are illustrated in (3.32), (3.33), and (3.34). Corresponding to the noun phrase with clausal postmodifier in each of these sentences is a sentential paraphrase that expresses the same relationships among the nominal and verbal elements involved. These are presented in Figure 3.5.

```
(3.32)  DECL     NP       DET      ADJ*     "the"
                          NOUN*    "machine"
                          PRPRTCL  VERB*    "coding"
                                   NP       DET      ADJ*     "these"
                                            AJP      ADJ*     "chemical"
                                            NOUN*    "structures"
                 VERB*    "is"
                 AJP      ADJ*     "fast"
                 PUNC     "."
```

```
        machine           coding
        structures        coding
        chemical          structures
```

```
(3.33) DECL    NP      DET      ADJ*     "the"
                       AJP      ADJ*     "chemical"
                       NOUN*    "structures"
                       PTPRTCL  VERB*    "coded"
                                PP       PREP     "by"
                                         DET      ADJ*     "this"
                                         NOUN*    "machine"
               VERB*   "are"
               AJP     ADJ*     "incorrect"
               PUNC    "."
```

```
       structures          coded
       chemical            structures
       machine             coded
```

```
(3.34) DECL    NP      PRON*    "they"
               VERB*   "designed"
               NP      NOUN*    "machines"
                       INFCL    INFTO    "to"
                                VERB*    "code"
                                NP       AJP      ADJ*     "chemical"
                                         NOUN*    "structures"
               PUNC    "."
```

```
       machines            code
       structures          code
       chemical            structures
```

In each of the sentences in Figure 3.5, the verb is the head of the construction, and the nominal arguments (whether noun phrases or prepositional phrases) are premodifiers and postmodifiers of the verb. This structure can be observed in (3.35), where the head of the sentence is the verb *coding*, which has *the machine* as a noun phrase premodifier, and *these chemical structures* as a noun phrase postmodifier.

Present participial clause as postmodifier:
the machine coding these chemical structures
Sentence:
the machine is coding these chemical structures

Past participial clause as postmodifier:
the chemical structures coded by this machine
Sentence:
the chemical structures were coded by this machine

Infinitival clause as postmodifier:
machines to code documents
Sentence:
machines code documents

FIGURE 3.5. Noun phrases with clausal postmodifiers and corresponding sentences.

```
(3.35)  DECL   NP      DET      ADJ*     "the"
                       NOUN*    "machine"
               VERB    "is"
               VERB*   "coding"
               NP      DET      ADJ*     "these"
                       AJP      ADJ*     "chemical"
                       NOUN*    "structures"
               PUNC    "."
```

In keeping with the general strategy of generating phrase descriptors that consist of a modifier and a head, these sentential constructions would yield descriptors having the nominal element as modifier and the verbal element as head. Given the close relationship in meaning between the sentential expressions and the nominal ones, phrase descriptors should be derived from them in such a way that they yield descriptors of similar form. By doing this, significant normalization is accomplished, since identical phrases are generated from constructions that differ syntactically but that are similar semantically.

The best way to assure that the nominal and verbal constructions are treated consistently is to simply transform the nominal constructions into verbal ones, and then apply the same phrase construction rules to them that would be applied to any other verbal construction. This is done by a rule that recognizes noun phrases with clausal postmodifiers, like the one in (3.32), and constructs the corresponding verb phrase, like the one in (3.36). Rules designed to operate on verb phrases then generate phrase descriptors consisting of verbal heads and nominal modifiers.

```
(3.36)  VP      NP      DET     ADJ*    "the"
                        NOUN*   "machine"
                VERB*   "coding"
                NP      DET     ADJ*    "these"
                        AJP     ADJ*    "chemical"
                        NOUN*   "structures"
```

An important beneficial aspect of this treatment of clausal postmodifiers of nouns is that the phrase descriptors generated are identical in form (after morphological regularization) to those that would be derived from corresponding nominal constructions that do not have clausal postmodifiers.[11] Examples appear in (3.37) and (3.38).

[11] **The phrase descriptors generated from these constructions are identical with the exception of *chemical coding* in (3.37). This inconsistency cannot be resolved on syntactic grounds. Though the desired phrase descriptor *chemical structure* is not constructed, the generated phrase *chemical coding* is not entirely inappropriate semantically.**

```
(3.37)  DECL    NP      NP      NOUN*   "machine"
                        AJP     ADJ*    "chemical"
                        NP      NOUN*   "structure"
                        NOUN*   "coding"
                VERB*   "is"
                AJP     ADJ*    "efficient"
                PUNC    "."
```

```
        machine             coding
        chemical            coding
        structure           coding
```

```
(3.38)  DECL    NP      NP      NOUN*   "machine"
                        NOUN*   "coding"
                        PP      PREP    "of"
                                AJP     ADJ*    "chemical"
                                NOUN*   "structures"
                VERB*   "is"
                AJP     ADJ*    "efficient"
                PUNC    "."
```

```
        machine             coding
        structures          coding
        chemical            structures
```

Figure 3.6 contains a list of verb bases that are excluded from use as elements of phrase descriptors.

accept	call	do	know	put
affect	change	exist	lead	receive
allow	come	expect	like	regard
appear	concern	express	make	see
approach	consider	give	meet	show
attempt	deal	have	need	suggest
base	describe	include	obtain	take
be	determine	intend	offer	use
become	devote	introduce	propose	
behave	discuss	involve	provide	

FIGURE 3.6. Semantically empty verbs.

3.4.6. Further Refinements

The basic phrase construction strategy, together with the treatment of conjoined modifiers and conjoined heads, succeeds in generating useful phrase descriptors from many constructions. With noun phrases of greater complexity, however, this strategy fails to identify many obviously appropriate phrase descriptors. In addition, some phrase descriptors of doubtful quality are constructed because the rules, as described so far, are not very selective in the choice of words to be included in phrase descriptors. The following two subsections discuss some approaches to alleviating these problems.

3.4.6.1. Replacement of Semantically General Heads with Modifiers

The noun phrase in (3.39) illustrates a common situation in which some good phrase descriptors are not generated by the basic phrase construction rules when they are applied to a complex noun phrase with several premodifiers.

```
(3.39)  NP      DET     ADJ*     "an"
                AJP     ADJ*     "automated"
                NP      NOUN*    "document"
                AJP     ADJ*     "clustering"
                NOUN*   "procedure"
                PUNC    "."
---------------------------------------------
        automated         procedure
        document          procedure
        clustering        procedure
```

It is well-known that substantial semantic information is required in order to produce an ideal analysis of complex nominals of this kind. In the absence of sufficient semantic information, it is not possible to determine the most appropriate internal structure of such noun phrases. Given this fact, the developer of a natural language processing system can choose either to generate all possible analyses, or to decide on a policy for selecting a single, reasonably useful analysis. The approach taken with the PLNLP grammar is essentially to identify the head of the noun phrase, and then attach all of the premodifiers at the same level, without any further substructure (except under certain special circumstances). The noun phrase in (3.39) is an example of this. Though this policy does not always yield the most desirable structure, it does provide an analysis that is a useful starting point for content analysis.

The simplified structure produced for noun phrases like the one in (3.39) accounts for the inability of the phrase indexing rules to construct the most desirable phrase descriptors from this noun phrase. Of the three phrase descriptors shown in (3.39), *automated procedure* and *clustering procedure* are appropriate. It would be preferable, however, to avoid constructing *document procedure*, and to generate *document clustering* instead. Also, *automated clustering* could be constructed in addition to *automated procedure*.

Though it is not possible to arrive at the ideal parse for a noun phrase of this kind using syntactic information alone, some progress can be made toward extracting more appropriate phrase descriptors from it by manipulating the parse tree in a simple way. An approach that has proven to be generally successful is to construct another related noun phrase by removing the current head (in this case *procedure*), and raising the nearest premodifier (here *clustering*) to the position of head. The result is the noun phrase in (3.40). From this derived noun phrase, the rules yield the desired phrase descriptors *automated clustering* and *document clustering*.

```
(3.40)  NP      DET     ADJ*     "an"
                AJP     ADJ*     "automated"
                NP      NOUN*    "document"
                NOUN*   "clustering"
                PUNC    "."
--------------------------------------------
        automated         clustering
        document          clustering
```

It would certainly not be appropriate to apply this strategy indiscriminately to all noun phrases. Rather, some conditions must be identified under which this strategy can be usefully applied. By examining a large number of complex noun phrases from which useful phrases can be generated in this way, it has been possible to identify a common characteristic shared by the heads of most of the noun phrases. In general, the heads of these constructions are nouns that refer to very general concepts, for example, *procedure*, *process*, *mechanism*, *technique*, *system*, and *approach*. Thus by defining a class of words with very general meanings, the conditions for this strategy can be specified (see Figure 3.7).[12]

ability	criteria	kind	procedure	structure
activity	criterion	manner	process	subject
amount	data	material	processing	system
analyses	design	matter	product	task
analysis	development	mean	production	technique
application	effect	mechanism	program	technology
approach	effort	method	property	theory
area	facility	methodology	purpose	topic
aspect	factor	model	result	totality
basis	field	operation	role	type
case	finding	pattern	scale	unit
category	form	plan	scheme	use
character	group	planning	series	value
characteristic	hypothesis	point	set	way
class	idea	practice	situation	
concept	issue	principle	solution	
consideration	item	problem	strategy	

FIGURE 3.7. General Nouns.

The examples in (3.41)-(3.44) illustrate this technique further. The original noun phrase in (3.41) yields the phrases *inverted approach* and *file approach*, which are of little value as content indicators. However, by constructing a related noun phrase by raising the premodifier *file* to the position of head, as in (3.42), the good phrase *inverted file* can be constructed. Similarly, the original noun phrase in (3.43) yields three phrases *conventional processes*, *information processes*, and *retrieval processes*. Though these phrases are not inaccurate indicators of document content, they are not ideal for use as descriptors, because they are not very specific in meaning. By removing the general noun *processes*, and raising *retrieval* to the position of

[12] The list of general nouns was initially compiled by examining the parsed text of the CISI query collection. This was then augmented by selecting additional high frequency words from the vocabulary of the entire CISI document collection.

head (as in (3.44)), two more specific phrases can be generated, *conventional retrieval* and *information retrieval.*

```
(3.41)  NP      AJP      ADJ*     "inverted"
                NP       NOUN*    "file"
                NOUN*    "approach"
                PUNC     "."
```

```
        inverted          approach
        file              approach
```

```
(3.42)  NP      AJP      ADJ*     "inverted"
                NOUN*    "file"
                PUNC     "."
```

```
        inverted          file
```

```
(3.43)  NP      AJP      ADJ*     "conventional"
                NP       NOUN*    "information"
                NP       NOUN*    "retrieval"
                NOUN*    "processes"
                PUNC     "."
```

```
        conventional      processes
        information       processes
        retrieval         processes
```

```
(3.44)  NP      AJP      ADJ*     "conventional"
                NP       NOUN*    "information"
                NOUN*    "retrieval"
                PUNC     "."
```

```
        conventional      retrieval
        information       retrieval
```

This strategy of raising a modifier to the position of head to replace a general head noun appears to be a generally useful technique for improving the quality of phrases. There are, however, situations in which undesirable phrases are produced. From the original phrase in (3.45), four phrase descriptors are produced: *large system*, *interactive system*, *document system*, and *retrieval system*. Except perhaps for the third one, all of these are acceptable for use as phrase descriptors. Two clearly desirable phrases, however, are missed: *interactive retrieval* and *document retrieval*. These phrases can be obtained by creating a related noun phrase with *retrieval* rather than *system* as head, as shown in (3.46). The unfortunate side effect of this, however, is that the undesirable phrase *large retrieval* is generated. Though a phrase such as this is not particularly valuable as a content indicator, it is not likely to have a strong negative influence on retrieval effectiveness, and thus can be accepted in order to obtain the clearly useful phrases *interactive retrieval* and *document retrieval*.

The strategy of replacing a general head by a modifier has further beneficial effects. Semantically general nouns are often essentially devoid of content themselves, but play an important role in providing links between related concepts in text.[13] Consider, for example, the noun phrase in (3.47). Application of the simple phrase construction procedure to this noun phrase would yield two phrase descriptors, *efficiency aspects* and *implementation aspects*. There is nothing particularly misleading about these phrases, but they are not ideal, because they fail to represent the essential equivalence in meaning of phrases like those in (3.48).

---

[13] Alternatively, these nouns could be called "semantically empty".

```
(3.45) NP      DET     ADJ*     "a"
               AJP     ADJ*     "large"
               AJP     ADJ*     "interactive"
               NP      NOUN*    "document"
               NP      NOUN*    "retrieval"
               NOUN*   "system"
               PUNC    "."
```

```
       large            system
       interactive      system
       document         system
       retrieval        system
```

```
(3.46) NP      DET     ADJ*     "a"
               AJP     ADJ*     "large"
               AJP     ADJ*     "interactive"
               NP      NOUN*    "document"
               NOUN*   "retrieval"
               PUNC    "."
```

```
       large            retrieval
       interactive      retrieval
       document         retrieval
```

```
(3.47) NP      DET     ADJ*     "the"
               NP      NOUN*    "efficiency"
               NOUN*   "aspects"
               PP      PREP     "of"
                       DET      ADJ*     "the"
                       NOUN*    "implementation"
               PUNC    "."
```

```
       efficiency       aspects
       implementation   aspects
```

(3.48) the efficiency aspects of the implementation
the efficiency of the implementation
implementation efficiency

The desired normalization can be accomplished by raising the premodifier *efficiency* in (3.47) to the position of head, replacing the semantically empty noun *aspects*. This yields the noun phrase in (3.49), from which the desired phrase descriptor *implementation efficiency* is constructed.

```
(3.49)  NP      DET      ADJ*     "the"
                NOUN*    "efficiency"
                PP       PREP     "of"
                         DET      ADJ*     "the"
                         NOUN*    "implementation"
                PUNC     "."
```

```
        implementation    efficiency
```

Constructions of this kind are common enough in text to make this manipulation useful, and the conditions for its application can be specified quite simply. Another example illustrating the value of this technique appears in (3.50).

```
(3.50)  NP      AJP      ADJ*     "possible"
                NP       NOUN*    "evaluation"
                NOUN*    "mechanisms"
                PP       PREP     "for"
                         NOUN*    "retrieval"
                         PP       PREP     "of"
                                  NOUN*    "documents"
                PUNC     "."
```

```
        evaluation        mechanisms
        retrieval         mechanisms
        documents         retrieval
```

Because the general noun *mechanisms* is the head of this noun phrase, the simplest form of the phrase construction method fails to produce the clearly desirable descriptor *retrieval evaluation*. However, by raising the premodifier *evaluation* to the position of head, as in (3.51), the desired phrase, *retrieval evaluation*, can be constructed.

```
(3.51)  NP      AJP     ADJ*    "possible"
                NOUN*   "evaluation"
                PP      PREP    "for"
                        NOUN*   "retrieval"
                        PP      PREP    "of"
                                NOUN*   "documents"
                PUNC    "."
```

```
        retrieval           evaluation
        documents           retrieval
```

The strategy of replacing these semantically empty heads with the nearest nominal modifier is based on the observation that the primary function of these lexical items, in the given syntactic context, is to provide a link between other words in text, or to express an idea in a more general manner. Removing the semantically general word has the effect of resolving an indirect syntactic relationship into a direct syntactic relationship. After doing this, the simple phrase construction method of associating heads with modifiers successfully generates more appropriate phrase descriptors.

The rules that perform the manipulation of syntactic structures and specify the conditions under which these manipulations can be performed can be viewed as a representation of knowledge about the function of common lexical items and expressions, and how to use this knowledge for extracting useful content indicators from text. It is important to emphasize that this knowledge is applicable to text in general. It is not directed toward the analysis of text from a narrow subject domain. This is the case, because the rules make use of common words and expressions that can be expected to occur in text of all kinds.

3.4.6.2. Exclusion of Semantically Empty Expressions

In much of the preceding discussion, a common method of avoiding the construction of phrase descriptors that are not good indicators of document content is to specify a list of words to be excluded from use as heads and/or modifiers in phrase descriptors. It is safe to take this approach, however, only if a broader context is not required to determine whether or not the word is likely to be a good content indicator. In other cases, it is necessary to consider the context of a word before it is possible to determine its value as a content indicator.

Some words are not good indicators of document content when used in one context, but are very important indicators of document content in other contexts. An example is the word *group*, which is not an important indicator of content in a phrase like *a group of scientists*, but is crucial to the meaning of a mathematical term like *group theory*. Thus it would not be advisable to simply exclude *group* from use as a phrase element. Nevertheless, it would be beneficial to avoid constructing phrases containing such words when they are not important to the meaning of the text. There are other nouns that have a function similar to *group* as it is used in the expression *a group of*. In general, this class can be characterized as having a collective meaning. Other members include *class*, *number*, and *totality*.

A similar situation arises with another class of words that can be described as having a generalizing function. For example, in a phrase such as *the concept of democracy*, *concept* is not essential to the meaning, whereas in a phrase like *the representation of abstract concepts*, it is more important.

And finally, there are words that serve primarily to express relationships among other elements of a text. For example, in an expression like *in terms of cost*, the word *terms* does not carry the essential meaning of the phrase, but in *the frequency of terms in the document*, it is important to the meaning of the phrase.

In cases like these, it is possible to quite confidently identify the occurrences that are not important indicators of content because they are used as parts of common expressions. The conditions on rules that exclude these expressions from being included in phrase descriptors have been written so that a variety of variants of an expression can be recognized and rejected without having to specify the exact form of each variant explicitly. For example, for expressions like those in (3.52), there is no need to specify the number of the head noun (*group*, *groups*), or the nature of its premodifiers.

(3.52) a large group of
a new group of
several groups of

Nevertheless, the entire expression can be excluded from the phrase indexing process.

All the criteria discussed in this section are used for specifying conditions under which words can be *excluded* from use as elements of phrase descriptors. It is important to take this approach, so that the indexing process remains applicable to all kinds of text. If the opposite approach is taken, the indexing process would have to make use of detailed information about the subject domain of the document collection in order to specify those terms that should be included as content indicators.

In order to increase its effectiveness, this approach to excluding semantically empty expressions from use as phrase elements would have to be extended greatly by compiling an extensive inventory of expressions that can be recognized and excluded. The examples discussed here are intended only to demonstrate the feasibility of the approach and its potential value. This approach could be used to exclude low-content occurrences of these words from use as single term descriptors, also.

3.4.6.3. Hyphenated forms

For purposes of identifying phrases for use as content indicators, hyphenated forms in the text of a document provide useful information, since they clearly indicate close relationships among words in the text. For example, given the hyphenated forms in text phrases such as those in (3.53), the phrases in (3.54) should certainly be used as phrase descriptors.

(3.53) machine-readable text
natural-language processing

(3.54)	**Modifier**	**Head**
	machine	readable
	natural	language

In order to provide maximum normalization of phrase descriptors, it is necessary to assure that text phrases containing hyphenated forms yield the same descriptors as related text phrases that do not contain hyphenated forms. That is, both phrases in (3.55), for example, should yield the same phrase descriptors. Because of the way the dictionary and grammar handle hyphenation, and the way that phrase descriptors are generated from noun phrases, this requires special processing.

(3.55) natural language processing
natural-language processing

The parse tree and phrase descriptors constructed by the indexing rules for the unhyphenated form appear in (3.56).

```
(3.56)  NP      AJP     ADJ*     "natural"
                NP      NOUN*    "language"
                NOUN*   "processing"
                PUNC    "."
```

natural	processing
language	processing
natural	language

From the parse tree representing the hyphenated form in (3.57), it can be seen that the hyphenated form is treated by the grammar as a single word, since it has the syntactic properties of its rightmost element. From the single phrase descriptor constructed from the hyphenated form by the indexing rules, a post-processing step constructs the second set of phrase descriptors shown in (3.57).

```
(3.57) NP      NP      NOUN*   "natural-language"
               NOUN*   "processing"
               PUNC    "."
```

Phrase descriptor constructed by indexing rules:
```
natural-language  processing
```

Phrase descriptors after post-processing:
```
natural           processing
language          processing
natural           language
```

By applying this post-processing step, comparable sets of phrase descriptors can be constructed from most hyphenated text forms and their unhyphenated counterparts.

The best way to deal with hyphenated forms would be to parse a group of words connected by hyphens separately from the rest of a sentence, and then use the resulting structure as a constituent in the structure representing the complete sentence. This would make it possible to include information about the internal structure of the hyphenated form in the final parse, and also avoid the post-processing step in phrase construction.

3.5. The Quality of Phrase Descriptors

Because the purpose of section 3.4 is to explain the approach to syntax-based phrase construction used in this study, the discussion concentrates on situations in which predominantly good phrase descriptors are constructed.

The purpose of this section is to examine the phrase construction process further in order to determine how successful the methods described in 3.4 are in consistently constructing semantically appropriate phrase descriptors. This is a way of subjectively assessing the quality of phrase descriptors, independent of their influence on retrieval effectiveness, which is discussed in section 3.6.

A phrase descriptor is held to be semantically appropriate if its meaning is closely related to the meaning of the text on which it is based, and if the desired normalization has been achieved. Two interdependent factors determine the semantic appropriateness of phrase descriptors: (1) the syntactic analysis system and (2) the phrase indexing procedure itself. The ability of the syntactic analysis system to correctly analyze the syntactic structure of a sentence, the treatment of syntactic ambiguity, and the treatment of parse failure all have a potentially strong influence on the quality of phrase descriptors. The ability of the phrase indexing rules to select good phrases, avoid useless phrases, decompose complex syntactic structures, and normalize semantically related constructions to a single form is also important. These issues are discussed in sections 3.5.1 and 3.5.2, and illustrated with representative examples. Section 3.5.3 provides some statistics related to parsing the document and query texts.

3.5.1. Problems Related Primarily to Syntactic Analysis

3.5.1.1. Syntactic Ambiguity

Syntactic ambiguity introduces situations in which the phrase indexing rules construct inappropriate phrase descriptors, or fail to construct desirable phrase descriptors. This section discusses two cases that illustrate the kinds of problems that arise when the phrase indexing rules are applied to syntactically ambiguous constructions.

A common and therefore important case of ambiguity arises when a present participle can be analyzed as either an adjectival premodifier of a noun, or as the head of a verbal construction. Examples appear in (3.58) and (3.59).

(3.58) They implemented a procedure for clustering documents.
They implemented a document clustering procedure.

(3.59) They design software for browsing interfaces.
They design browsing interface software.

As shown by the trees in (3.60) and (3.61), the sentences in (3.58) yield two parses each.[14]

---

[14] In these diagrams, parse trees are presented in the order determined by the parse metric. In multiple parses, the P-METRIC value appears just below each parse tree. According to the evaluation strategy used by the parse metric, a smaller value for P-METRIC is preferred over a higher value. The parses trees are thus presented in order from lowest to highest P-METRIC. When referring to the ranking of parses according to the parse metric, phrases such as "first parse", "second parse", etc., are used. The phrases "best parse", and "preferred parse" are used to refer to the most appropriate parse, as determined by the meaning of the sentence. Though the parse metric often succeeds in identifying the best parse and presenting it first, it is not correct in all cases.

(3.60a) They implemented a procedure for clustering documents.

```
DECL    NP      PRON*   "they"
        VERB*   "implemented"
        NP      DET     ADJ*    "a"
                NOUN*   "procedure"
                PP      PREP    "for"
                        AJP     ADJ*    "clustering"
                        NOUN*   "documents"
        PUNC    "."
P-METRIC = 0.221
```

```
*documents          procedure
*clustering         documents
```

(3.60b)

```
DECL    NP      PRON*   "they"
        VERB*   "implemented"
        NP      DET     ADJ*    "a"
                NOUN*   "procedure"
                PP      PREP    "for"
                        VERB*   "clustering"
                        NP      NOUN*   "documents"
        PUNC    "."
P-METRIC = 0.221
```

```
clustering          procedure
documents           clustering
```

```
(3.61a) They implemented a document clustering procedure.
        DECL    NP      PRON*   "they"
                VERB*   "implemented"
                NP      DET     ADJ*    "a"
                        NOUN*   "document"
                        PRPRTCL VERB*   "clustering"
                                NP      NOUN*   "procedure"
                PUNC    "."
        P-METRIC = 0.221
```

```
        document            clustering
        *procedure          clustering
```

```
(3.61b) DECL    NP      PRON*   "they"
                VERB*   "implemented"
                NP      DET     ADJ*    "a"
                        NP      NOUN*   "document"
                        AJP     ADJ*    "clustering"
                        NOUN*   "procedure"
                PUNC    "."
        P-METRIC = 0.23
```

```
        *document           procedure
        clustering          procedure
        document            clustering
```

Ideally, the phrase descriptors in (3.62) would be constructed from both of these sentences, since these descriptors correctly decompose and normalize the more complex noun phrases to the same forms as are constructed from the simpler text phrases shown in (3.63).

(3.62)

Modifier	**Head**
clustering	procedure
document	clustering

(3.63)

Text Phrases	**Phrase Descriptors**
clustering procedure	clustering procedure
procedure for clustering	
document clustering	document clustering
clustering of documents	

Due to the ambiguity of these noun phrases, and the parse metric that ranks alternative parses, the desired descriptors appear in the second parse for both of these sentences. These are the preferred parses for these sentences. The phrase descriptors marked with an asterisk in (3.60) and (3.61) are inappropriate because they do not correspond to the desired normalized forms shown in (3.62). Of the two preferred parses, only (3.60b) yields exactly the desired descriptors. In addition to the correct descriptors, the less desirable phrase *document procedure* appears in (3.61b).

It would be possible to write rules that would construct the desired phrases from the trees in (3.60a) and (3.61a). However, this would result in inappropriate phrase descriptors in other situations in which the first parse is preferred. For example, from the sentences in (3.59), the ideal phrase descriptors would be those shown in (3.64), since they are identical to those derived from the simpler text phrases shown in (3.65).

(3.64)

Modifier	Head
interface	software
browsing	interface
browsing	software

(3.65)

Text Phrases	Phrase Descriptors
interface software	interface software
software for interfaces	
browsing interface	browsing interface
interface for browsing	
browsing software	browsing software
software for browsing	

As can be seen from trees (3.66) and (3.67), no single parse for these sentences yields all of the desired phrase descriptors. The first parse, (3.66a), is best, but it yields only two of the three desired phrases; *browsing software* is lacking. The second parse, (3.66b), yields one good descriptor and one inappropriate descriptor.

```
(3.66a)  They design software for browsing interfaces.
         DECL     NP       PRON*    "they"
                  VERB*    "design"
                  NP       NOUN*    "software"
                           PP       PREP     "for"
                                    AJP      ADJ*     "browsing"
                                    NOUN*    "interfaces"
                  PUNC     "."
         P-METRIC = 0.211
---------------------------------------------------------------
         interfaces         software
         browsing           interfaces
---------------------------------------------------------------
(3.66b)  DECL     NP       PRON*    "they"
                  VERB*    "design"
                  NP       NOUN*    "software"
                           PP       PREP     "for"
                                    VERB*    "browsing"
                                    NP       NOUN*    "interfaces"
                  PUNC     "."
         P-METRIC = 0.211
---------------------------------------------------------------
         browsing           software
         *interfaces        browsing
```

The parser succeeds in producing a single parse for the sentence in (3.67), but again only two of the three desired descriptors are constructed. Because *interface* does not get raised to the position of head, *browsing interface* is not constructed (for clarification, see section 3.4.6.1).

```
(3.67)  They design browsing interface software.
        DECL    NP      PRON*   "they"
                VERB*   "design"
                NP      AJP     ADJ*    "browsing"
                        NP      NOUN*   "interface"
                        NOUN*   "software"
                PUNC    "."
```

```
        browsing            software
        interface           software
```

Since this kind of ambiguity cannot be solved on syntactic grounds, and since the verbal analysis is preferred in some cases, and the adjectival analysis is preferred in others, it is clear that the phrase indexing strategy should be designed to get as many good descriptors as possible from both types of construction. Under these circumstances, it appears that the best that can be done is to use all of the phrase descriptors for the ambiguous constructions in (3.60), (3.61), and (3.66). This is not an ideal solution to the problem, since at least one inappropriate descriptor will be constructed for each sentence. This approach does, however, assure that all of the appropriate descriptors are generated, which assures that the phrase descriptors generated from the sentences in (3.58) and (3.59) will have an opportunity to match on descriptors generated from the simpler phrases in (3.63) and (3.65).

Conjunctions are another source of ambiguity. In some situations, the phrase construction process and the multiple parses resulting from this source of ambiguity interact in such a way that there is little negative effect on the phrase descriptors generated for a particular construction. In other situations, ambiguity related to conjunctions introduces rather serious problems for this approach to phrase indexing.

The noun phrase in (3.68) is an example in which ambiguity has a relatively slight impact on the quality of phrase descriptors constructed. The best parse for this noun phrase, (3.68b), is ranked second by the parse metric. In this analysis, the head *system* has a single premodifying noun phrase, *information storage and retrieval.* In the first parse ((3.68a)), however, *system* has two noun phrase premodifiers, *information*, and *storage and retrieval.*

Though these structures differ significantly, the phrase indexing rules succeed in constructing nearly identical phrase descriptors from both of them. The descriptors are the same except for *information system*, which appears in (3.68a) but not (3.68b). However, since this phrase is semantically appropriate, nothing is lost by using phrase descriptors from the first parse, rather than the second parse, which is actually the best analysis. Examples of this kind are evidence that syntactic ambiguity does not always result in serious inaccuracies when constructing phrase descriptors.

(3.68a) The information storage and retrieval system.

```
NP      DET     ADJ*    "the"
        NP      NOUN*   "information"
        NP      NP      NOUN*   "storage"
                CONJ*   "and"
                NP      NOUN*   "retrieval"
        NOUN*   "system"
        PUNC    "."
P-METRIC = 0.7
```

```
information       system
storage           system
retrieval         system
information       storage
information       retrieval
```

(3.68b)

```
NP      DET     ADJ*    "the"
        NP      NP      NOUN*   "information"
                NP      NOUN*   "storage"
                CONJ*   "and"
                NP      NOUN*   "retrieval"
        NOUN*   "system"
        PUNC    "."
P-METRIC = 0.8
```

```
storage           system
retrieval         system
information       storage
information       retrieval
```

(3.68c)

```
NP      DET     ADJ*    "the"
        NP      NP      NOUN*   "information"
                NOUN*   "storage"
        CONJ*   "and"
        NP      NP      NOUN*   "retrieval"
                NOUN*   "system"
        PUNC    "."
P-METRIC = 6.4
```

```
information       storage
retrieval         system
```

In other cases where conjunctions lead to ambiguity, much more serious difficulties may arise. The sentence in (3.69) is an example.

(3.69) Titles are important not only in commercial services, such as Chemical Titles, BASIC, Current Contents, CA Condensates, but also in scanning primary journals, and in traditional library services, such as bibliographies.

This sentence yields six parses, the first of which appears in (3.70). The best parse, (3.71), is unfortunately ranked lowest by the parse metric, receiving a rank of six.

All nine phrase descriptors constructed for the best parse are semantically appropriate, and good indicators of content. Of the 16 descriptors constructed from the first parse, seven are desired descriptors that also appear in the best parse. The other nine, however, are less appropriate descriptors that are not ideal indicators of content.

Taking all six parses into consideration, seven of the nine good phrase descriptors constructed for the best parse were constructed from all six of the parses. Of the remaining two good phrase descriptors, one was constructed for five of the parses, and the other was constructed for three of the parses. The phrase construction rules are thus quite successful in that they identified a high proportion of the desired descriptors in all parses. However, syntactic ambiguity has a strong negative effect on the quality of phrase descriptors in this case, since in addition to seven good phrase descriptors, the first parse is assigned nine additional undesirable phrases descriptors.

These examples thus illustrate the range of effects that syntactic ambiguity can have on the quality of phrase descriptors. The effect is slight in cases like (3.68), but can be quite strong in more complex constructions like (3.69).

```
(3.70)  ...
PP      CONJ    "not only"
        PP      PREP    "in"
                AJP     ADJ*    "commercial"
                NOUN*   "services"
        CONJ    ","
        PP      PREP    "such as"
                NP      NP      NP      NOUN*   "Chemical"
                                NOUN*   "Titles"
                        CONJ    ","
                        NP      NOUN*   "BASIC"
                        CONJ    ","
                        NP      NP      NOUN*   "Current"
                                NOUN*   "Contents"
                        CONJ*   ", and"
                        NP      NP      NOUN*   "CA"
                                NOUN*   "Condensates"
                                PUNC    ","
                                PP      AVP     AVP     ADV*    "but"
                                                ADV*    "also"
                                        PREP    "in"
                                        AJP     ADJ*    "scanning"
                                        NOUN*   "primary"
                NOUN*   "journals"
        CONJ*   ", and"
        PP      PREP    "in"
                AJP     ADJ*    "traditional"
                NP      NOUN*   "library"
                NOUN*   "services"
                PUNC    ","
                PP      PREP    "such as"
                        NOUN*   "bibliographies"

        P-METRIC = 0.32568
```

commercial	services	*Titles	journals
Chemical	Titles	*BASIC	journals
Current	Contents	*Contents	journals
CA	Condensates	*Condensates	journals
traditional	services	*primary	Titles
library	services	*primary	BASIC
bibliographies	services	*primary	Contents
		*primary	Condensates
		*scanning	primary

```
(3.71)  ...
PP      CONJ     "not only"
        PP       PREP     "in"
                 AJP      ADJ*     "commercial"
                 NOUN*    "services"
        CONJ     ","
        PP       PREP     "such as"
                 NP       NP       NOUN*    "Chemical"
                          NOUN*    "Titles"
                 CONJ     ","
                 NP       NOUN*    "BASIC"
                 CONJ     ","
                 NP       NP       NOUN*    "Current"
                          NOUN*    "Contents"
                 CONJ*    ", and"
                 NP       NP       NOUN*    "CA"
                          NOUN*    "Condensates"
        CONJ*    ", but also"
        PP       PP       PREP     "in"
                          VERB*    "scanning"
                          NP       AJP      ADJ*     "primary"
                                   NOUN*    "journals"
                 CONJ*    ",       and"
                 PP       PREP     "in"
                          AJP      ADJ*     "traditional"
                          NP       NOUN*    "library"
                          NOUN*    "services"
                          PUNC     ","
                          PP       PREP     "such as"
                                   NOUN*    "bibliographies"

        P-METRIC = 0.596
```

commercial	services
Chemical	Titles
Current	Contents
CA	Condensates
journals	scanning
primary	journals
traditional	services
library	services
bibliographies	services

3.5.1.2. Failed Parses

One of the features of the PLNLP system that makes it well-suited to analysis of unrestricted text is its ability to deal in a useful way with ungrammatical strings. If a string cannot be parsed successfully, the parse fitting procedure is invoked (see section 3.3). This procedure attempts to identify useful lower-level constituents in the string. While this strategy does provide some useful constructions for the phrase indexing rules to operate on, some inaccuracies are also introduced.

The string in (3.72) is a series of comma-separated noun phrases rather than a sentence. Since by default the parser expects input strings to be sentences, it attempts to analyze this string as a sentence. This attempt fails, however, since it is not a sentence. The parse fitting procedure then takes over and generates the constructions appearing in (3.72).

Like the normal parsing operation, the fitting procedure is by default oriented toward recognizing verbal constructions. Thus the VP with head *direct* and the VERB *list* are identified in preference to the possible noun phrase analyses. This, of course, causes the phrase construction rules to fail to identify several good descriptors, for example, *direct access*, *data base*, *inverted list*, and *threaded list*. In addition, some inappropriate phrases are constructed: *time inverted*, and *list threaded*. In spite of these shortcomings, the descriptors *information retrieval* and *access time* are good content indicators.

Another way in which a parse can fail is if the input string is actually a sentence, but it cannot be recognized as a sentence because it does not have a structure defined by the grammar rules. This is the case for the sentence in (3.73), where the parenthesized abbreviation, *(WLN)*, could not be attached as a postmodifier of the noun *Notation.* Nevertheless, the preceding portion of the sentence was correctly recognized as a verb phrase (actually a complete sentence), from which four good phrase descriptors are constructed.

(3.72) Information retrieval, direct access memory, disk, drum, data base, access time, inverted list, threaded list.

```
XXXX    NP      NP      NOUN*   "information"
                NOUN*   "retrieval"
                PUNC    ","
        VP*     VERB*   "direct"
                NP      NP      NOUN*   "access"
                        NOUN*   "memory"
                        PUNC    ","
                        NAPPOS  NOUN*   "disk"
                                PUNC    ","
        NP      NOUN*   "drum"
                PUNC    ","
                NAPPOS  NOUN*   "data"
        NP      NOUN*   "base"
                PUNC    ","
        NP      NP      NOUN*   "access"
                NOUN*   "time"
                PTPRTCL PUNC    ","
                        VERB*   "inverted"
        NP      NOUN*   "list"
                PTPRTCL PUNC    ","
                        VERB*   "threaded"
        VERB    "list"
        PUNC    "."
```

```
information       retrieval
access            memory
*time             inverted
access            time
*list             threaded
```

```
(3.73)    XXXX      VP*       NP        DET       ADJ*      "this"
                                        NOUN*     "paper"
                              VERB*     "deals"
                              PP        PREP      "with"
                                        DET       ADJ*      "the"
                                        NP        NOUN*     "encoding"
                                        CONJ*     "and"
                                        NP        NOUN*     "decoding"
                                                  PP        PREP      "of"
                                                            DET       ADJ*      "a"
                                                            NP        NOUN*     "Wiswesser"
                                                            NP        NOUN*     "Line"
                                                            NOUN*     "Notation"
                    NP        PUNC      "("
                              NOUN*     "WLN"
                              PUNC      ")"
                    PUNC      "."
```

Notation	encoding
Notation	decoding
Wiswesser	Notation
Line	Notation

3.5.1.3. Other Parsing Problems

Two other characteristics of the syntactic analysis system can have a negative effect on the quality of phrase descriptors, these are (a) the strategy for analyzing the structure of complex noun phrases, and (b) the policy used for prepositional phrase attachment.

The internal structure of noun phrases with several premodifiers, like the one in (3.74), cannot be correctly determined on the basis of syntactic information alone. Rather than constructing all possible parses, in most cases, this syntactic analysis system simply identifies the head of the noun phrase and attaches all of the premodifiers at the same level. Given this structure, the strategy of associating the head with each modifier yields the three phrases shown in (3.74). Of these phrases, only *file organization* is clearly a good indicator of content. In addition, the procedure fails to construct the desirable descriptor *direct access*.

```
(3.74)  NP      AJP     ADJ*    "direct"
                NP      NOUN*   "access"
                NP      NOUN*   "file"
                NOUN*   "organization"
-----------------------------------------
        *direct          organization
        *access          organization
        file             organization
```

Though the strategy of raising a premodifier to the position of head, as discussed in section 3.4.6.1, helps to lessen the negative effects of this problem for many noun phrases, it does not help in cases such as this. As a result, when applied to some complex noun phrases, the phrase construction rules fail to construct some desirable phrase descriptors, and in addition construct some inappropriate ones.

The noun phrase in (3.75) illustrates a situation in which an undesirable phrase descriptor is constructed because a prepositional phrase is attached as a modifier of the wrong noun. In this tree, the prepositional phrase *in preparation of sdi profiles* is attached as a modifier of *analyzer*, when ideally it should modify *use*. Given only the syntactic information available to the parser, the best position for attachment of prepositional phrases cannot be determined, so they are simply attached as modifiers of the nearest available noun. This will not always yield the correct analysis, but this policy is preferable to generating a profusion of parses, and does provide an analysis that is often useful for further processing. In this case, however, the undesirable phrase descriptor *preparation analyzer* is constructed.

Usually, incorrect prepositional phrase attachment results in the construction of only a few poor descriptors, and may also cause the phrase construction procedure to fail to identify a few good phrases. However, this problem can interact in complex ways with other aspects of a parse and lead to the construction of a number of poor phrase descriptors. The noun phrase in (3.76), for example, shows what can happen when the ambiguity of conjunctions interacts with prepositional phrase attachment. The useless phrase descriptors shown in (3.76) are constructed, and some good descriptors are missed, for example, *industrial firms.*

```
(3.75)  NP      NOUN*   "use"
                PP      PREP    "of"
                        DET     ADJ*    "an"
                        AJP     ADJ*    "automatic"
                        NP      NOUN*   "text"
                        NOUN*   "analyzer"
                        PP      PREP    "in"
                                NOUN*   "preparation"
                                PP      PREP    "of"
                                        NP      NOUN*   "sdi"
                                        NOUN*   "profiles"
                PUNC    "."
```

```
        automatic       analyzer
        text            analyzer
        *preparation    analyzer
        profiles        preparation
        sdi             profiles
```

(3.76) the politicians in Albany or Sacramento, in Washington, Paris, or Moscow, the managers of far-flung industrial firms, or the people who run educational institutions

Modifier	**Head**
*Washington	Albany
*Paris	Albany
*Moscow	Albany
*Washington	Sacramento
*Paris	Sacramento
*Moscow	Sacramento

3.5.2. Problems Related Primarily to the Phrase Construction Method

As currently implemented, the phrase construction rules successfully identify semantically appropriate phrases and accomplish useful normalization when applied to a variety of syntactic construction types. There are, however, constructions that the rules do not handle adequately, so the quality of phrase descriptors is adversely affected. The inadequacies in the phrase indexing rules that are responsible for production of less desirable phrase descriptors are of two kinds. First, there are problems that can be solved quite easily by relatively minor modifications of the phrase indexing rules. Second, there are problems that can very likely be solved to a large extent, but the solutions will require quite substantial extensions of the existing phrase indexing strategy.

The first kind of problem has to do with excluding words and common expressions from use as elements of phrase descriptors. Section 3.4.2.1 discussed ways of preventing certain classes of single words from being used as heads and modifiers in phrase descriptors. Section 3.4.6.2 extended this strategy by excluding semantically empty expressions that may consist of more than a single word. A substantial number of phrase descriptors that are clearly useless as content indicators could be avoided if this general idea could be developed further.

An example is the expression *as a function of*, as it appears in (3.77). This expression gives rise to three phrase descriptors that have no direct relationship to the meaning of the text, and that could potentially match with phrase descriptors drawn from texts that differ significantly in meaning from the source of these phrases.

A few additional examples, together with the undesirable phrase descriptors generated from them, are shown in (3.78). Expressions of this kind can be identified in a straightforward fashion by expanding the rules discussed in section 3.4.6.2. However, in order for this approach to have a substantial effect on overall phrase quality, a comprehensive inventory of expressions such as these would have to be made. An inventory of expressions for this purpose should be based on a systematic study of a large sample of the vocabulary and text of titles and abstracts from a variety of scientific and technical fields.

```
(3.77)  PP      PREP    "as"
                DET     ADJ*    "a"
                NOUN*   "function"
                PP      PREP    "of"
                        NP      DET     ADJ*    "the"
                                NP      NOUN*   "data"
                                NOUN*   "base"
                        CONJ    ","
                        NP      DET     ADJ*    "the"
                                NOUN*   "demands"
                                PP      PREP    "on"
                                        PRON*   "it"
                        CONJ*   ",      and"
                        NP      DET     ADJ*    "a"
                                NOUN*   "parameter"
                                RELCL   NP      PRON*   "which"
                                        NP      DET     ADJ*    "the"
                                                NP      NOUN*   "system"
                                                NOUN*   "designer"
                                        VERB    "may"
                                        VERB*   "control"
```

```
        *base           function
        *demands        function
        *parameter      function
        data            base
        system          designer
```

(3.78) a screening process *in conjunction with* other extracting techniques
*conjunction process
*techniques conjunction

a *multitude of* other sources of references
*sources multitude

the *majority of* names of carbon compounds
*name majority

a good deal of turmoil
*good deal
*turmoil deal

a social phenomenon *in its own right*
*right phenomenon

The second kind of problem has to do with the more complex task of identifying relationships among words in text. The purpose of decomposing syntactic constructions by combining a head with its modifiers is to construct phrases descriptors that represent relationships between words, even though the words may not be contiguous in the text. A simple extension to this basic strategy was discussed in section 3.4.6.1, which makes it possible to recognize even more indirect relationships between words. Using that approach, it is possible, for example, to construct the phrase descriptor *implementation efficiency* from the text phrase *the efficiency aspects of the implementation*, even though there is no direct syntactic relationship between *efficiency* and *implementation*.

This is a useful refinement, but it is not capable of dealing adequately with constructions like the noun phrase in (3.79). Six of the phrase descriptors constructed from this noun phrase are not good indicators of document content. It would be a simple matter to have the phrase indexing rules prevent the word *area* from being included in a phrase descriptor when it occurs in the expression *in such areas as*. This would avoid the undesirable phrase descriptors, but is not a completely satisfying solution.

```
(3.79)  NP      NOUN*   "applications"
                PP      PREP    "of"
                        NOUN*   "automation"
                        PP      PREP    "in"
                                AJP     ADJ*    "such"
                                NOUN*   "areas"
                                PP      PREP    "as"
                                        NP      NOUN*   "circulation"
                                        CONJ    ","
                                        NP      NOUN*   "cataloging"
                                        CONJ    ","
                                        NP      NOUN*   "acquisitions"
                                        CONJ    ","
                                        NP      AJP     ADJ*    "serial"
                                                NOUN*   "records"
                                        CONJ*   ",      and"
                                        NP      AJP     ADJ*    "other"
                                                NOUN*   "record-keeping"
```

```
        automation         applications
        *areas             automation
        *circulation       areas
        *cataloging        areas
        *acquisitions      areas
        *records           areas
        *record-keeping    areas
        serial             records
```

Much more benefit could be gained by recognizing that the prepositional phrase *in such areas* establishs a link between the noun *automation* and the prepositional phrase *as circulation, cataloging, acquisitions, serial records, and record-keeping*. Once this relationship is recognized, the noun phrase object of the preposition *as*, can be raised to the position of the head noun *areas*, as shown in (3.80).

```
(3.80)  NP      NOUN*   "applications"
                PP      PREP    "of"
                        NOUN*   "automation"
                        PP      PREP    "in"
                                NP      NOUN*   "circulation"
                                CONJ    ","
                                NP      NOUN*   "cataloging"
                                CONJ    ","
                                NP      NOUN*   "acquisitions"
                                CONJ    ","
                                NP      AJP     ADJ*    "serial"
                                        NOUN*   "records"
                                CONJ*   ",      and"
                                NP      AJP     ADJ*    "other"
                                        NOUN*   "record-keeping"
```

automation	applications
circulation	automation
cataloging	automation
acquisitions	automation
records	automation
record-keeping	automation
serial	records

This manipulation resolves an indirect syntactic relationship into a direct one, and thus makes it possible to produce the more appropriate phrase descriptors shown in (3.80) by application of existing phrase indexing rules.

It is likely that many additional expressions could be identified that have functions similar to that of *in such areas as*. Compiling an inventory of such expressions and determining how each one should be treated in order to yield the most appropriate phrase descriptors will be a project of quite large proportions. This would constitute a substantive addition to the phrase indexing process that should yield significant improvements.

The syntactic relationship of modification between the head of a construction and its modifiers is the basis for all of the phrase construction rules presently implemented. There are, of course, other kinds of syntactic constructions that should be exploited as sources of phrase descriptors. It is important to attempt to extend the phrase indexing method in order to deal with such constructions, since they provide further opportunities for identifying relationships among words in text and normalizing the form of expressions that are semantically similar but syntactically different.

Given the sentences in (3.81), the rules discussed in section 3.4 make it possible to recognize a relationship between *query* and *analysis* (in their various forms), and to construct phrase descriptors that normalize the various text forms to a single form (after morphological regularization). This is possible because the words involved enter into a direct syntactic relationship of modification with one another. The same semantic relationship between *query* and *analysis* that is expressed in these sentences, however, can be expressed using constructs in which the relationship is expressed indirectly.

For example, the sentence in (3.82) expresses the same semantic relationship between *query* and *analysis* that is expressed by the sentences in (3.81). But here, the relationship is expressed indirectly via the verb *submitting*, rather than directly by a syntactic relationship of modification. That is, in (3.81), the semantic relationship has a direct syntactic correlate: *query* (in one of its forms) modifies *analysis* (in one of its forms). In (3.82), however, the semantic relationship is expressed indirectly, since *queries* and *analysis* are both modifiers of *submitting*, instead of *queries* being a modifier of *analysis*. In this expression, the verb *submit* conveys no real content independently of its arguments (modifiers). Its function is to indicate that a relationship exists between its arguments. Given the syntactic analysis in (3.82), the desired phrase descriptor could be constructed by associating the head of the NP postmodifier of *submitting* with the head of a PP postmodifier that has *to* as its preposition.

(3.81)	**Sentence**	**Phrase Descriptor**	
	They designed a query analysis system.	query	analysis
	They designed a system for analyzing the queries.	queries	analyzing
	The system analyzing the queries is automatic.	queries	analyzing
	The queries analyzed by the system are well-constructed.	queries	analyzed
	They designed a system to analyze queries automatically.	queries	analyze

Several verbs other than *submit* also have the function of expressing a relationship of this kind between their complements, for example, *subject X to Y*, *perform X on Y*. By incorporating this kind of information into the phrase construction process, significant improvements could be made in the quality of normalization and the number of phrase descriptors generated.[15]

[15] The approach to lexicography developed by Mel'chuk, Apresjan, and Zholkovsky may be useful in developing this extension to the phrase construction process, as their concept of *lexical function* provides a mechanism for formalizing this kind of paraphrase relationship

(3.82) This research is about submitting user's queries to automatic text analysis.

```
DECL    NP      DET     ADJ*     "this"
                NOUN*   "research"
        VERB*   "is"
        PP      PREP    "about"
                VERB*   "submitting"
                NP      NP       NOUN*    "user"
                                 POSSESS "'s"
                        NOUN*    "queries"
                PP      PREP     "to"
                        AJP      ADJ*     "automatic"
                        NP       NOUN*    "text"
                        NOUN*    "analysis"
        PUNC    "."
```

Before this approach can be implemented, however, another problem must be overcome. The syntactic structure assigned by the PLNLP grammar to the sentence in (3.82) is as shown in (3.83). The difference is that in (3.83) the prepositional phrase *to automatic text analysis* is a postmodifier of the noun *queries*, rather than of the verb *submitting*. This is not the correct analysis for the sentence, but is explained by the policy for prepositional phrase attachment, and the fact that the grammar does not make extensive use of information about the complement structure of verbs. For purposes of phrase indexing, this problem can be solved either by altering the grammar so that it produces the analysis shown in (3.82), or by writing the phrase construction rules in such a way that information about the function of verbs like *submit*, as well as knowledge about the nature of the analysis assigned to

(Apresjan, Mel'chuk, and Zholkovsky 1969; Mel'chuk and Zholkovsky 1970; Zholkovsky and Mel'chuk 1970; Mel'chuk 1981).

these constructions by the grammar, would both be taken into consideration. Either of these alternatives would be possible, but improving the grammar would be preferable.

Note that in (3.83), there is a direct syntactic relationship between *queries* and *analysis*, and that a phrase descriptor containing both of these words is constructed. The desired normalization is not achieved, however, since the elements of the phrase are not in the same order as they are in the phrases in (3.81).

(3.83) This research is about submitting user's queries to automatic text analysis.

```
DECL     NP       DET      ADJ*     "this"
                  NOUN*    "research"
         VERB*    "is"
         PP       PREP     "about"
                  VERB*    "submitting"
                  NP       NP       NOUN*    "user"
                                    POSSESS  "'s"
                           NOUN*    "queries"
                           PP       PREP     "to"
                                    AJP      ADJ*     "automatic"
                                    NP       NOUN*    "text"
                                    NOUN*    "analysis"
         PUNC     "."
```

```
queries            submitting
user               queries
*analysis          queries
automatic          analysis
text               analysis
automatic          text
```

3.5.3. Parsing the Document and Query Collections

The first step in preparing the document and query collections for parsing is to break the text into strings that are bounded by punctuation that is characteristic of sentence boundaries.[16] This segmentation is done automatically with a fairly high degree of accuracy. However, before parsing, the collections were examined visually in order to catch any obvious errors made by the program that separates running text into sentences.

By default, the parser attempts to analyze each input string as a grammatically correct sentence. An attractive feature of the PLNLP system, however, is that it is also possible to instruct the parser to attempt to analyze an input string as some other grammatical construction, for example, a noun phrase. This feature has an important application in text analysis for document retrieval, namely, in analyzing document titles. All titles in the experimental collections used in this study were parsed in this way. In rare cases, a title may actually be a complete sentence, or perhaps a prepositional phrase. The vast majority of titles are noun phrases, however, so it is beneficial to use this feature in analyzing them. Since titles are easily recognizable automatically in the original text of the document collections, this feature can be invoked automatically.

---

[16] It is sometimes convenient to call these strings sentences, even though they may be noun phrases (for example, document titles), or other strings that are not grammatically correct sentences.

This facility for parsing noun phrases can also be used in analyzing queries. Rather than being statements of information need expressed in complete sentences, a query may be a sequence of noun phrases separated by commas, semi-colons or periods. In order to examine the importance of this characteristic of query statements, the query collections were parsed in two different ways. First, they were prepared for parsing exactly as document collections are. Second, each query was examined manually, and portions having the form of noun phrases, rather than sentences, were identified so that the parser would attempt to analyze them as noun phrases rather than as sentences. Retrieval experiments were then done using queries parsed in both ways.

There are three possible results from parsing an input string.

(1) If there are no cases of syntactic ambiguity in the string, and it has a structure defined by the grammar, then a single parse results. This will be a sentence, in the default case, or a noun phrase if the facility for recognizing noun phrases is applied, as for titles.

(2) If the string cannot be recognizing as a grammatically correct construction, then the fitting procedure is invoked. The result is a single fitted parse containing lower-level constructions that were identified during the attempted parse. Note that parse failure can arise in two situations: (a) the input string may be grammatically incorrect, in which case a successful parse would not be expected, or (b) the string may be grammatically correct, but it does not correspond to a syntactic structure that is defined by the grammar.

(3) If the string is grammatically correct, but syntactically ambiguous, more than one parse will result. A limit of ten has been placed on the number of parses displayed in cases of ambiguity, but all parses are actually completed.

In each of these cases, the phrase indexing rules are applied to all of the structures produced by the syntactic analysis system.

3.5.3.1. Parsing Statistics

Some summary statistics related to parsing the document and query collections appear in Tables 3.1-3.4. Tables 3.1 and 3.2 show the minimum, maximum, and mean number of words per sentence, and the minimum, maximum, and mean CPU seconds per sentence required to parse the collections. CPU times are for execution on an IBM 3090. Tables 3.3 and 3.4 contain figures showing the number of parses produced for sentences from the two collections. Part (a) of Table 3.4 for example, shows that of the 8805 sentences in the document collection, 3680 (about 41%) of them yielded a single parse, and 2951 (33%) resulted in fitted parses. Parts (b) and (c) of the table contain similar statistics for the query collection, parsed both without special attention to noun phrases (part (b)), and with queries consisting of noun phrases parsed as noun phrases rather than as sentences (part (c)).

	CACM Document Collection 3204 Documents, 10,111 Sentences		
	Minimum	Maximum	Mean
Words per Sentence	1	90	16.88
CPU Seconds per Sentence	0.10	669.04	6.26

(a)

	CACM Query Collection Sentence Parse 52 Queries, 88 Sentences		
	Minimum	Maximum	Mean
Words per Sentence	2	33	12.53
CPU Seconds per Sentence	0.19	17.00	3.03

(b)

	CACM Query Collection Sentence and Noun Phrase Parse 52 Queries, 105 Sentences		
	Minimum	Maximum	Mean
Words per Sentence	1	29	10.73
CPU Seconds per Sentence	0.10	17.23	2.56

(c)

TABLE 3.1. Sentence length and CPU time for parsing, CACM collection.

	CISI Document Collection 1460 Documents, 8805 Sentences		
	Minimum	Maximum	Mean
Words per Sentence	1	88	21.05
CPU Seconds per Sentence	0.11	658.47	10.31

(a)

	CISI Query Collection Sentence Parse 76 Queries, 268 Sentences		
	Minimum	Maximum	Mean
Words per Sentence	3	77	17.68
CPU Seconds per Sentence	0.34	93.06	6.47

(b)

	CISI Query Collection Sentence and Noun Phrase Parse 76 Queries, 270 Sentences		
	Minimum	Maximum	Mean
Words per Sentence	2	77	17.54
CPU Seconds per Sentence	0.19	93.71	6.49

(c)

TABLE 3.2. Sentence length and CPU time for parsing, CISI collection.

CACM							
	10,111 Sentences; Maximum parses: 174						
Number of Parses	0 Fitted	1	2	3	4-5	6-10	>10
Number of Sentences	2627	4790	1761	362	335	154	82
Fraction of Sentences	0.260	0.474	0.174	0.036	0.033	0.015	0.008
(a) Documents (3204)							
	88 Sentences; Maximum parses: 12						
Number of Parses	0 Fitted	1	2	3	4-5	6-10	>10
Number of Sentences	53	20	8	1	5	0	1
Fraction of Sentences	0.602	0.227	0.091	0.011	0.057	0	0.011
(b) Queries (52); Sentence Parse							
	105 Sentences; Maximum parses: 21						
Number of Parses	0 Fitted	1	2	3	4-5	6-10	>10
Number of Sentences	28	50	17	2	6	0	2
Fraction of Sentences	0.267	0.476	0.162	0.019	0.057	0	0.019
(c) Queries (52); Sentence and Noun Phrase Parse							

TABLE 3.3. Parsing statistics for the CACM collection.

CISI							
	8805 Sentences; Maximum parses: 48						
Number of Parses	0 Fitted	1	2	3	4-5	6-10	>10
Number of Sentences	2951	3680	1198	284	357	244	91
Fraction of Sentences	0.335	0.418	0.136	0.032	0.040	0.028	0.010
(a) Documents (1460)							
	268 Sentences; Maximum parses 22						
Number of Parses	0 Fitted	1	2	3	4-5	6-10	>10
Number of Sentences	70	127	42	9	15	3	2
Fraction of Sentences	0.261	0.474	0.156	0.034	0.056	0.011	0.007
(b) Queries (76); Sentence Parse							
	270 Sentences; Maximum parses: 22						
Number of Parses	0 Fitted	1	2	3	4-5	6-10	>10
Number of Sentences	56	143	40	10	16	3	2
Fraction of Sentences	0.207	0.529	0.148	0.037	0.059	0.011	0.007
(c) Queries (76); Sentence and Noun Phrase Parse							

TABLE 3.4. Parsing statistics for the CISI collection.

3.6. Retrieval Experiments

The retrieval experiments discussed in this section have been carried out in order to determine the influence of syntax-based phrase descriptors on retrieval effectiveness. Two experimental collections have been used, CISI and CACM. These collections were chosen because they lie at the extremes of the retrieval performance spectrum when indexed with non-syntactic phrases. As shown by the experiments discussed in chapter 2, significant improvements in retrieval effectiveness can be achieved when non-syntactic phrases are used with CACM, but only a slight increase is possible with CISI. The performance of syntactic phrase indexing on these two collections therefore provides a reasonable indication of the relative effectiveness of syntactic and non-syntactic phrase descriptors.

Section 3.6.1 explains how the phrases constructed using the syntax-based method described in section 3.4 are incorporated into document and query vectors. Section 3.6.2 defines four parameters that determine the content of phrase subvectors and control the effect of phrase matches on query-document similarity values. Section 3.6.3 presents the results of the retrieval experiments, and explains how the parameters introduced in section 3.6.2 influence retrieval effectiveness. In chapter 4, these syntax-based phrase indexing results are compared to the non-syntactic results of chapter 2, as well as to previous experimental work on phrase indexing.

3.6.1. Construction of Document and Query Vectors

In order to fairly evaluate the relative effectiveness of the syntactic and non-syntactic phrase indexing methods examined in this study, the document and query vectors used in the comparative experiments should differ only with respect to the basis on which the phrase descriptors are constructed. All other characteristics of the vectors must remain constant. This requirement must be maintained so that any differences in retrieval performance can be unequivocally attributed to differences in the phrase construction method. Thus for each collection used in both the syntactic and non-syntactic retrieval experiments, the single term subvectors are identical, all phrase descriptors contain two elements, the same stemming operation is applied to single terms and phrase elements, and the same term weighting functions are used.

The construction of a document vector containing syntactic phrases is illustrated by the following example. Figure 3.8 contains the original text of document 175 from the CACM collection. From this, the syntactic analyzer and phrase construction rules yield the parse tree and phrases in Figure 3.9. A stemming procedure is then applied to the phrase elements. From the original document text and these stemmed phrases, standard software from the SMART package is used to construct the final document vector shown in Figure 3.10. The single term and phrase weights in this vector were calculated according to the functions defined in section 2.2.3.

The solution of simultaneous ordinary differential equations using a general purpose digital computer.

FIGURE 3.8. Text of CACM document 175.

```
NP      DET     ADJ*    "the"
        NOUN*   "solution"
        PP      PREP    "of"
                AJP     ADJ*    "simultaneous"
                AJP     ADJ*    "ordinary"
                AJP     ADJ*    "differential"
                NOUN*   "equations"
                PRPRTCL VERB*   "using"
                        NP      DET     ADJ*    "a"
                                NP      AJP     ADJ*    "general"
                                        NOUN*   "purpose"
                                AJP     ADJ*    "digital"
                                NOUN*   "computer"
        PUNC    "."
```

```
equations        solution
simultaneous     equations
ordinary         equations
differential     equations
purpose          computer
digital          computer
general          purpose
```

FIGURE 3.9. Parse tree for CACM document 175, with syntactic phrases.

Document Number	Descriptor Number	Weight	Descriptor Type	Descriptor
175	4111	0.2202	0	gener
175	12651	0.3373	0	digit
175	27890	0.4248	0	simultan
175	29560	0.2481	0	solut
175	29565	0.4480	0	ordin
175	41114	0.3313	0	purpos
175	41155	0.1333	0	comput
175	47336	0.2978	0	equ
175	47831	0.4228	0	differ
175	5227	0.3603	1	differ equ
175	14464	0.2353	1	digit comput
175	15528	0.2758	1	gener purpos
175	23969	0.2323	1	purpos comput
175	30239	0.3729	1	ordin equ
175	30536	0.3613	1	simultan equ
175	51984	0.2729	1	equ solut

FIGURE 3.10. Weighted vector for CACM document 175 containing single term and syntactic phrase descriptors.

3.6.2. Syntactic Phrase Indexing and Retrieval Parameters

Experiments were done to examine the effects of four parameters that determine (a) the content of the phrase subvector, and (b) the influence that phrase matches have on the similarity value calculated for a query-document pair. Two of these parameters are related to syntactic analysis; these are the *parse threshold* and the *query parsing mode*. The others, *df-phrase* and the *phrase subvector weight*, are related to the frequency characteristics of phrase descriptors, and to weighting.

Parse Threshold. In cases where the parser yields more than one parse for sentences involving syntactic ambiguity, the phrase construction rules generate phrase descriptors from all parses up to a maximum of ten. As explained in section 3.5.1.1, the best parse for a sentence is not always the one ranked highest by the parse metric. In such cases, some good phrase descriptors may be lost if only phrases from the first parse are included in the document or query vector. The parse threshold parameter specifies a limit on the number of parses from which phrase descriptors are taken. With a parse threshold of one, only phrase descriptors from the first parse would be included in a vector. With a threshold of two, phrase descriptors from the first two parses would be used, and so on.

In order to examine the importance of phrase descriptors from multiple parses, retrieval experiments were done with four different versions of the document and query collections. These versions were constructed by using parse thresholds of 1, 2, 5, and 10.

Query Parsing Mode. Queries are not always stated as complete sentences. Instead, a query may be stated as a single complex noun phrase, or as a sequence of noun phrases separated by punctuation. For purposes of constructing phrase descriptors, it is advantageous to have the parser analyze such strings as noun phrases rather than as sentences. As explained in section 3.5.4, the PLNLP system provides this capability.

In order to examine the importance of analyzing queries in this way, the query collections were parsed using two parsing modes. With the default sentence parsing mode, the parser attempts to analyze each input string as a sentence. Using the noun phrase parsing mode, each query that consists of a noun phrase or sequence of noun phrases was identified so that the parser would attempt to analyze it as a noun phrase rather than as a sentence. Queries that are to be analyzed as noun phrases must be identified manually.

The example in (3.84) indicates the potential value of parsing such strings as noun phrases rather than using the default sentence mode. Analyzed as a noun phrase, the string yields the phrase *image processing*, which is the focus of the query. When analyzed as a sentence, however, this phrase is not constructed. It should be noted that even in default sentence parsing mode, many strings that are not sentences are correctly analyzed as noun phrases due to the parse fitting procedure (see sections 3.3.1 and 3.3.2). The example in (3.84) shows that the ideal result is not always achieved, however.

Retrieval experiments were done with the query collections parsed in both the default sentence mode and the noun phrase mode.

Phrase Subvector Weight. If the phrase descriptors constructed on the basis of syntactic information are predominantly good indicators of document content, it might be expected that retrieval performance could be enhanced by increasing the importance of matches between query and document phrases. This possibility was tested by examining the effect of increasing the contribution that a phrase match makes to the similarity value calculated for a query-document pair.

(3.84a) Image recognition and any other methods of automatically transforming printed text into computer-ready form.

NOUN PHRASE PARSE

```
NP  NP      NP      NOUN*   "image"
            NOUN*   "recognition"
    CONJ*   "and"
    NP      QUANT   ADJ*    "any"
            QUANT   ADJ*    "other"
            NOUN*   "methods"
            PP      PREP    "of"
                    AVP     ADV*    "automatically"
                    VERB*   "transforming"
                    NP      AJP     ADJ*    "printed"
                            NOUN*   "text"
                            PP      PREP    "into"
                                    AJP     ADJ*    "computer-ready"
                                    NOUN*   "form"
    PUNC    "."
```

image	recognition	text	transforming
transforming	recognition	printed	text
transforming	methods	form	text
automatically	transforming	computer-ready	form

(3.84b) SENTENCE PARSE

```
IMPR  VERB*   "image"
      NP      NP      NOUN*   "recognition"
              CONJ*   "and"
              NP      QUANT   ADJ*    "any"
                      QUANT   ADJ*    "other"
                      NOUN*   "methods"
                      PP      PREP    "of"
                              AVP     ADV*    "automatically"
                              VERB*   "transforming"
                              NP      AJP     ADJ*    "printed"
                                      NOUN*   "text"
                                      PP      PREP    "into"
                                              AJP     ADJ*    "computer-ready"
                                              NOUN*   "form"
      PUNC    "."
```

transforming	recognition	printed	text
transforming	methods	form	text
automatically	transforming	computer-ready	form
text	transforming		

Using the similarity function defined in section 2.2.3.3, the weight of a phrase match can be increased by specifying a phrase subvector weight that is greater than one. The overall similarity between query vector q and document vector d is calculated as a weighted sum of the innerproduct similarity values calculated for the single term and phrase subvectors; see expression (3.85).

$$sim(q,\ d) = (c_s \cdot ip(q_s,\ d_s)) + (c_p \cdot ip(q_p,\ d_p)) \tag{3.85}$$

Here, c_s and c_p are weights applying to the single term and phrase subvectors, respectively. In these experiments, 1.00 has been used for c_s, and 1.00, 1.25, 1.50, and 2.00 have been tested as values of the phrase subvector weight, c_p.

Document Frequency of Phrases (df-phrase). As defined in the discussion of non-syntactic phrase indexing in chapter 2 (see section 2.2.1), the parameter df-phrase is a threshold used to place restrictions on the document frequency of phrase descriptors that are included in document and query vectors. The experiments with non-syntactic phrases showed that retrieval effectiveness can generally be increased slightly by excluding phrase descriptors that have relatively high document frequencies. The effect of a continuum of document frequency thresholds has been tested for syntactic phrases, also.

3.6.3. Retrieval Results

The best retrieval results that have been achieved using the syntax-based phrase construction strategy are summarized in Table 3.5. Retrieval effectiveness is expressed as percent change in average precision in comparison to simple single term indexing. Also in this table are the values for the parameters introduced in section 3.6.2 that yielded these results. Table 3.6 contains the corresponding complete recall and precision results.

These figures show that for the CACM collection, a rather modest increase in average precision of 8.7% is attained. This increase is statistically significant, but cannot be characterized as "material" according to the criteria suggested by Sparck Jones (1974).[17] When applied to the CISI collection, syntactic phrase indexing yields only a very slight increase in average precision of 1.2%. This increase is neither statistically significant nor material.

These retrieval results are discussed further in chapter 4. The remainder of this section describes the way in which the parameters defined in section 3.6.2 affect retrieval performance.

[17] The Wilcoxon signed rank test for paired observations was used to determine the statistical significance of the changes in average precision.

Coll.	Parameters				Avg. Prec.	Stat. Signif.
	Parse Thresh.	Query Parsing Mode	Phrase Sub-vector Weight	Phrase Doc. Freq. (df-phrase)	Change	Change?
CACM	1	Noun Phrase	1.25	< 40 0.0125n	+8.7%	yes P < 0.01
CISI	1	Noun Phrase	1.00	< 20 0.0137n	+1.2%	no P > 0.18

TABLE 3.5. Summary of best retrieval results for syntactic phrase indexing. Average precision is with respect to single term indexing, see Table 3.6. The value n is collection size; for CACM, $n = 3204$, for CISI, $n = 1460$.

Recall Level	Precision			
	CACM		CISI	
	Single Term Indexing	Syntactic Phrase Indexing	Single Term Indexing	Syntactic Phrase Indexing
0.10	0.5086	0.5636	0.4919	0.4932
0.20	0.4343	0.4728	0.4032	0.4041
0.30	0.3672	0.4318	0.3118	0.3208
0.40	0.2972	0.3261	0.2624	0.2680
0.50	0.2398	0.2550	0.2320	0.2326
0.60	0.1912	0.2010	0.1901	0.1935
0.70	0.1462	0.1486	0.1504	0.1553
0.80	0.1086	0.1088	0.1119	0.1094
0.90	0.0711	0.0694	0.0739	0.0756
1.00	0.0610	0.0579	0.0521	0.0518
Avg Prec	0.2604	0.2830	0.2450	0.2480
% Change		8.7		1.2

TABLE 3.6. Average precision at 10 recall levels for single term and syntactic phrase indexing applied to the CACM and CISI collections.

Base parameter values. Part (a) of Table 3.7 contains average precision figures for syntactic phrase indexing using a set of base parameter values. These figures provide a point of reference for analyzing the effects of other parameter values.

Using a parse threshold of one, the default query parsing mode, a phrase subvector weight of 1.00, and no document frequency restrictions, CACM yields a 5.8% increase in average precision, and CISI yields a 1.0% decrease in average precision. These figures indicate that the syntactic phrases generated for CACM are predominantly good indicators of document content that have a positive influence on retrieval performance. For CISI, however, the small decrease in average precision suggests that the syntax-based phrase construction process yields a mixture of good and bad phrase descriptors. Averaged over the experimental query collection, the positive and negative effects of these good and bad descriptors tend to neutralize one another, yielding a net effect that is small and negative.

The effect of df-phrase. A continuum of values for df-phrase were tested on both collections in order to determine the optimal value for this parameter. The values that yield the largest increases in average precision over the base values are given in part (b) of Table 3.7. The effect of excluding high document frequency syntactic phrases is similar to the effect noted for non-syntactic phrases. That is, removal of high document frequency phrases has a small, positive influence on retrieval effectiveness. For CACM, the increase is a very slight 0.3% over syntactic phrases with no document frequency restrictions. For CISI, exclusion of phrases having a document frequency greater than 20 yields an increase of 1.0% in average precision over simple single term indexing, which is a difference of two percentage points over syntactic phrases with no document frequency restrictions.

The effect of phrase subvector weighting. Part (c) of Table 3.7 shows the level of retrieval effectiveness that can be achieved when the best values for the df-phrase and phrase subvector weight parameters are used. For CACM, increasing the phrase subvector weight to 1.25 increases the average precision change to 7.2% over simple single term indexing, which is a 1.1% increase over a phrase subvector weight of 1.00. Weights above 1.25 yield poorer average precision for CACM. This is an indication that even though phrase descriptors tend to have an overall positive effect on the CACM collection, if they are given too much weight, they begin to overshadow the effects of the single term descriptors, which also play an important role in retrieval effectiveness.

Any increase in the phrase subvector weight for the CISI collection results in worse performance than the default weight of 1.00. This result is further support for the earlier observation that a substantial proportion of the phrase descriptors assigned to the CISI collection are not good indicators of document content, and therefore have a negative effect on retrieval performance.

In addition to experimenting with increased phrase subvector weights, reduced weights of 0.50 and 0.75 were also tested on both CACM and CISI. For both collections, reduced weights yielded slightly poorer performance than the optimal weights given in Table 3.5.

The effect of the parse threshold. If the parse threshold is increased above one, so that phrases from parses in addition to the first one are included as phrase descriptors, the effect on both collections is that average precision decreases slightly below the levels shown in Table 3.7 (c). The negative effect of using phrases from additional parses indicates that phrases taken from parses of lower rank tend to be less appropriate as indicators of document content than phrases taken from parses of higher rank. This in turn suggests that the parse metric that provides a ranking of multiple parses (see section 3.3.1) tends to provide useful information about the probable appropriateness of alternative parses.

The small effect that increasing the parse threshold has on retrieval performance is most likely due to the fact that a relatively small proportion of sentences have more than one parse (see Tables 3.3 and 3.4, section 3.5.4.1). In addition, even when a sentence has a relatively large number alternative parses, the number of new phrases constructed from parses after the first one is often small. An example of this situation appears in section 3.5.1.1.

The effect of the query parsing mode. By comparing the average precision figures in part (c) of Table 3.7 with those in Table 3.5, it can be seen that using the noun phrase query parsing mode has a small positive effect for both collections. For CACM the noun phrase mode increases average precision change to +8.7% from the +7.2% that results when default parsing mode is used. For CISI, the difference is even smaller, increasing to +1.2% from +1.0%.

Collection	Parameters				Average
	Parse Threshold	Query Parsing Mode	Phrase Subvector Weight	Phrase Document Frequency (df-phrase)	Precision
CACM	1	Sentence (default)	1.00	none	+5.8% 0.2754
CISI	1	Sentence (default)	1.00	none	-1.0% 0.2426

(a)
Base parameter values.

CACM	1	Sentence (default)	1.00	< 40	+6.1% 0.2764
CISI	1	Sentence (default)	1.00	< 20	+1.0% 0.2473

(b)
Base parameter values plus best df-phrase.

CACM	1	Sentence (default)	1.25	< 40	+7.2% 0.2790
CISI	1	Sentence (default)	1.00	< 20	+1.0% 0.2473

(c)
Base parameter values plus best df-phrase and best phrase subvector weight.

TABLE 3.7. Average precision for various parameter values. Percent change is with respect to single term indexing (CACM: 0.2604; CISI: 0.2450; see also Table 3.6).

CHAPTER 4

COMPARISON OF PHRASE INDEXING EXPERIMENTS

4.1. Syntactic vs. Non-syntactic Phrase Indexing

The results of retrieval experiments comparing the effectiveness of single term indexing and phrase indexing were discussed briefly in sections 2.3 and 3.6.[1] This section discusses in more detail the relative effectiveness of single term indexing and phrase indexing, as well as the relative effectiveness of syntactic and non-syntactic phrase indexing when applied to the CACM and CISI collections.

Table 4.1 exhibits the results of retrieval experiments comparing single term indexing to both syntactic and non-syntactic phrase indexing. The results in Table 4.2 compare non-syntactic and syntactic phrase indexing directly. The results in these tables were obtained using the best phrase indexing and retrieval methods discussed in chapters 2 and 3.

In comparing phrase indexing with single term indexing, these figures show that only non-syntactic phrase indexing applied to the CACM collection yields a material increase in average precision, a 22.7% increase over single

[1] Unless further clarification is given, the phrases "syntactic phrase indexing" and "non-syntactic phrase indexing" are used in this chapter to refer to the phrase indexing methods presented in chapters 2 and 3.

Recall Level	CACM			CISI		
	Single	Phrase Indexing		Single	Phrase Indexing	
	Terms	Non-syntactic	Syntactic	Terms	Non-syntactic	Syntactic
0.10	0.5086	0.6489	0.5636	0.4919	0.4947	0.4932
0.20	0.4343	0.5335	0.4728	0.4032	0.4026	0.4041
0.30	0.3672	0.4542	0.4318	0.3118	0.3285	0.3208
0.40	0.2972	0.3569	0.3261	0.2624	0.2712	0.2680
0.50	0.2398	0.2971	0.2550	0.2320	0.2330	0.2326
0.60	0.1912	0.2416	0.2010	0.1901	0.1982	0.1935
0.70	0.1462	0.1719	0.1486	0.1504	0.1556	0.1553
0.80	0.1086	0.1261	0.1088	0.1119	0.1131	0.1094
0.90	0.0711	0.0742	0.0694	0.0739	0.0811	0.0756
1.00	0.0610	0.0615	0.0579	0.0521	0.0582	0.0518
Avg Prec	0.2604	0.3195	0.2830	0.2450	0.2503	0.2480
% Change		22.7	8.7		2.2	1.2

TABLE 4.1. Average precision at 10 recall levels for single term indexing and phrase indexing.

Recall Level	CACM		CISI	
	Phrase Indexing		Phrase Indexing	
	Syntactic	Non-syntactic	Syntactic	Non-syntactic
0.10	0.5636	0.6489	0.4932	0.4947
0.20	0.4728	0.5335	0.4041	0.4026
0.30	0.4318	0.4542	0.3208	0.3285
0.40	0.3261	0.3569	0.2680	0.2712
0.50	0.2550	0.2971	0.2326	0.2330
0.60	0.2010	0.2416	0.1935	0.1982
0.70	0.1486	0.1719	0.1553	0.1556
0.80	0.1088	0.1261	0.1094	0.1131
0.90	0.0694	0.0742	0.0756	0.0811
1.00	0.0579	0.0615	0.0518	0.0582
Avg Prec	0.2830	0.3195	0.2480	0.2503
% Change		12.9		0.9

TABLE 4.2. Average precision at 10 recall levels for syntactic and non-syntactic phrase indexing.

term indexing. Syntactic phrase indexing results in an increase of only 8.7% over single term indexing for CACM. The increases in average precision due to phrase indexing on the CISI collection are clearly insignificant for both the non-syntactic (2.2%) and syntactic (1.2%) methods.

In comparing syntactic and non-syntactic phrase indexing, the precision figures in Table 4.2 show that on the average, the non-syntactic method yields better results than the syntax-based method for both collections. For CACM, the 12.9% increase in average precision is both material and statistically significant.[2] The increase of 0.9% for the CISI collection, is insignificant, however.

A prominent characteristic of these results is the small overall effect that phrase indexing appears to have on retrieval effectiveness. Two factors appear to explain this small effect. The first factor is the number of phrase descriptors that occur in both document and query vectors. The second factor is that average precision figures are derived by averaging the performance of an entire collection of queries. This may tend to obscure significant variation in the performance of individual queries.

4.1.1. The Number of Query Phrases Occurring in Documents

Phrase indexing cannot have a strong influence on retrieval effectiveness unless there is the potential for frequent phrase matches between queries and

---

[2] $P < 0.01$, Wilcoxon signed rank test for paired observations.

documents. In order to provide for frequent phrase matches, phrases that occur in queries must be assigned frequently as phrase descriptors in documents. The statistics in Tables 4.3 and 4.4 help to explain why syntax-based phrase indexing has such a small influence on retrieval effectiveness for both CACM and CISI, and also why non-syntactic phrase indexing has a small effect on CISI.

The line labeled "All Syntactic Phrases" in Table 4.3 shows that when all phrases identified by the syntax-based phrase construction method are assigned as phrase descriptors, 2937, or 92%, of the 3204 documents in the CACM collection contain at least one phrase descriptor. On average, each document contains about 15 phrase descriptors, which accounts for 37% of the descriptors in each document. However, since only a small fraction of these phrases also occur in queries, only a relatively small proportion of them can match query phrases, and thus contribute to the similarity between a query and document. The next line in the table shows how many of the phrases in the documents also occur in queries. Only 715, or 22%, of the documents contain phrases that also occur in queries. Averaged over the collection, this is less than one phrase per document. When restrictions on the document frequency of phrases are applied, these figures are reduced further. When phrases with document frequencies of 40 or greater are excluded, only 604, or 19% of the documents contain phrases. This is the syntactic phrase indexing method that yields the best retrieval results.

CACM				
Phrase Indexing Method	Number of Vectors with Phrases	Mean Single Terms per Vector	Mean Phrases per Vector	Mean Ratio of Phrases to All Descriptors
Documents (3204)				
All Syntactic Phrases	2937 92%	20.22	15.54	0.37
Syntactic Query Phrases	715 22%	20.22	0.35	0.01
Syntactic Query Phrases (df < 40)	604 19%	20.22	0.27	0.01
Non-syntactic Query Phrases (df < 90)	2072 65%	20.22	9.22	0.17
Queries (52)				
All Syntactic Phrases	50 96%	10.67	3.79	0.27
Syntactic Phrases (df < 40)	50 96%	10.67	3.79	0.27
Non-syntactic Phrases (df < 90)	52 100%	10.67	47.37	0.70

TABLE 4.3. Statistics on phrase descriptors in CACM documents and queries.

CISI				
Phrase Indexing Method	Number of Vectors with Phrases	Mean Single Terms per Vector	Mean Phrases per Vector	Mean Ratio of Phrases to All Descriptors
Documents (1460)				
All Syntactic Phrases	1460 100%	45.20	34.43	0.43
Syntactic Query Phrases	1124 77%	45.20	2.31	0.05
Syntactic Query Phrases (df < 20)	896 61%	45.20	1.27	0.03
Non-syntactic Query Phrases (df < 30)	1280 88%	45.20	3.39	0.07
Queries (76)				
All Syntactic Phrases	74 97%	22.59	8.58	0.29
Syntactic Phrases (df < 20)	72 95%	22.59	6.97	0.24
Non-syntactic Phrases (df < 30)	76 100%	22.59	11.22	0.32

TABLE 4.4. Statistics on phrase descriptors in CISI documents and queries.

In contrast to these figures for syntactic phrase indexing, using the best non-syntactic phrase indexing method, 2072, or 65%, of the documents in the CACM collection contain phrase descriptors that also occur in queries. On average, this is about nine phrase descriptors per document, or 17% of the descriptors in each document. The far greater number of phrase descriptors assigned as a result of non-syntactic phrase indexing is accounted for by the much less selective nature of the non-syntactic method, and the highly unrestrictive parameter values that yield the best retrieval results for CACM. Given these figures, it is to be expected that the non-syntactic method, which assigns about nine query phrases to each document, would have a stronger influence on retrieval effectiveness than the syntactic method, which assigns less than one query phrase to each document.

The corresponding statistics for the CISI collection appear in Table 4.4. Here, the best syntactic phrase indexing method assigns query phrases to 896, or 61%, of the documents in the collection. This averages out to fewer than two phrases per document, or about 3% of the descriptors in each document. The best non-syntactic method assigns phrases to 1280, or 88%, of the documents. This is an average of fewer than 4 phrases per document, or about 7% of the descriptors in each document. The difference in the number of phrase descriptors assigned by the non-syntactic method for CACM and CISI is accounted for by the more restrictive parameter values used for CISI.

The difference between the average proportion of phrase descriptors assigned by the syntactic and non-syntactic methods to the documents of the CISI collection (3% vs. 7%) is much smaller than that for the CACM collection (1% vs. 17%). In addition, the average proportion of phrase descriptors assigned by both the syntactic and non-syntactic phrase indexing methods for CISI (3% and 7%) is closer to the proportion assigned by the syntactic method for CACM (1%) than to the proportion assigned by the non-syntactic method for CACM (17%). These statistics account, in part, for the relatively small influence that syntactic phrase indexing has on retrieval effectiveness for both collections, as well as the small effect that non-syntactic phrase indexing has on retrieval effectiveness for the CISI collection.

4.1.2. The Performance of Individual Queries

Average precision figures such as those in Tables 4.1 and 4.2 are calculated by figuring the average precision for each query over recall levels 0.10 through 0.90, and then averaging this value over the entire query collection. Evaluation measures of this kind are useful as general indicators of the relative effectiveness of different indexing and retrieval strategies. It is also, instructive, however, to examine the performance of individual queries. This makes it possible to determine whether or not different indexing and retrieval strategies perform consistently on most queries.

Tables 4.5 and 4.6 contain average precision figures for each query in the CACM and CISI collections.

CACM						
Query	Average Precision			Percent Change in Average Precision (b - a) / a		
	(1) Single Term Indexing (ST)	(2) Phrase Indexing Non-syntactic (Nsyn.)	(3) Phrase Indexing Syntactic (Syn.)	(4) (a) ST vs. (b) Nsyn.	(5) (a) ST vs. (b) Syn.	(6) (a) Syn. vs. (b) Nsyn.
1	0.1707	0.2829	0.2015	65.73	18.04	40.40
2	0.0018	0.0018	0.0018	0.00	0.00	0.00
3	0.0803	0.1381	0.1390	71.98	73.10	-0.65
4	0.0572	0.0737	0.0683	28.85	19.41	7.91
5	0.1509	0.1286	0.1984	-14.78	31.48	-35.18
6	0.2049	0.4101	0.2049	100.15	0.00	100.15
7	0.1851	0.2663	0.1991	43.87	7.56	33.75
8	0.1240	0.1361	0.3200	9.76	*158.06*	-57.47
9	0.1146	0.1306	0.1187	13.96	3.58	10.03
10	0.6905	0.7624	0.6264	10.41	-9.28	21.71
11	0.3764	0.4374	0.4106	16.21	9.09	6.53
12	0.2441	0.4463	0.3404	82.83	39.45	31.11
13	0.2778	0.2963	0.3603	6.66	29.70	-17.76
14	0.4898	0.4143	0.5021	-15.41	2.51	-17.49
15	0.1871	0.2290	0.1656	22.39	-11.49	38.28
16	0.0638	0.0687	0.0623	7.68	-2.35	10.27
17	0.1209	0.1641	0.1384	35.73	14.47	18.57
18	0.0955	0.0840	0.1719	-12.04	80.00	-51.13
19	0.3695	0.5110	0.4554	38.29	23.25	12.21
20	0.0938	0.5390	0.1358	*474.63*	44.78	*296.91*
21	0.0611	0.1794	0.0621	193.62	1.64	188.89
22	0.6145	0.6650	0.6197	8.22	0.85	7.31
23	0.0589	0.0511	0.0737	-13.24	25.13	-30.66
24	0.1056	0.1151	0.1048	9.00	-0.76	9.83
25	0.2258	0.2790	0.2539	23.56	12.44	9.89
26	0.3871	0.4859	0.4103	25.52	5.99	18.43

TABLE 4.5 (a). Average precision for each CACM query.

CACM						
Query	Average Precision			Percent Change in Average Precision (b - a) / a		
	(1) Single Term Indexing (ST)	(2) Phrase Indexing: Non-syntactic (Nsyn.)	(3) Phrase Indexing: Syntactic (Syn.)	(4) (a) ST vs. (b) Nsyn.	(5) (a) ST vs. (b) Syn.	(6) (a) Syn. vs. (b) Nsyn.
27	0.2673	0.3226	0.2524	20.69	-5.57	27.81
28	0.5375	0.7453	0.5375	38.66	0.00	38.66
29	0.6507	0.7201	0.6499	10.67	-0.12	10.80
30	0.1679	0.3566	0.2089	112.39	24.42	70.70
31	0.8431	0.7176	0.7176	-14.89	-14.89	0.00
32	0.4077	0.7255	0.2591	77.95	**-36.45**	180.01
33	0.0833	0.0909	0.0833	9.12	0.00	9.12
36	0.2659	0.3536	0.2843	32.98	6.92	24.38
37	0.2178	0.2307	0.1880	5.92	-13.68	22.71
38	0.3466	0.3852	0.4763	11.14	37.42	-19.13
39	0.3308	0.2919	0.3383	-11.76	2.27	-13.72
40	0.3140	0.3416	0.5198	8.79	65.54	-34.28
42	0.0480	0.0908	0.0480	89.17	0.00	89.17
43	0.2249	0.2471	0.2249	9.87	0.00	9.87
44	0.0274	0.0324	0.0267	18.25	-2.55	21.35
45	0.2692	0.3213	0.2633	19.35	-2.19	22.03
48	0.0961	0.0289	0.1704	**-69.93**	77.32	**-83.04**
49	0.1196	0.1980	0.1114	65.55	-6.86	77.74
57	1.0000	1.0000	1.0000	0.00	0.00	0.00
58	0.1988	0.2511	0.2599	26.31	30.73	-3.39
59	0.3877	0.3745	0.3656	-3.40	-5.70	2.43
60	0.2974	0.2290	0.2958	-23.00	-0.54	-22.58
61	0.2421	0.3635	0.2402	50.14	-0.78	51.33
62	0.0782	0.0796	0.0742	1.79	-5.12	7.28
63	0.4415	0.7079	0.6488	60.34	46.95	9.11
64	0.1250	0.1111	0.1250	-11.12	0.00	-11.12

TABLE 4.5 (b). Average precision for each CACM query.

CISI						
	Average Precision			Percent Change in Average Precision (b - a) / a		
Query	(1) Single Term Indexing (ST)	(2) Phrase Indexing Non-syntactic (Nsyn.)	(3) Phrase Indexing Syntactic (Syn.)	(4) (a) ST vs. (b) Nsyn.	(5) (a) ST vs. (b) Syn.	(6) (a) Syn. vs. (b) Nsyn.
1	0.5120	0.4941	0.4946	-3.50	-3.40	-0.10
2	0.0318	0.0321	0.0318	0.94	0.00	0.94
3	0.2300	0.2300	0.2300	0.00	0.00	0.00
4	0.0598	0.0545	0.0590	-8.86	-1.34	-7.63
5	0.0518	0.0511	0.0512	-1.35	-1.16	-0.20
6	0.0233	0.0222	0.0233	-4.72	0.00	-4.72
7	0.0184	0.0167	0.0241	-9.24	30.98	**-30.71**
8	0.0467	0.0450	0.0462	-3.64	-1.07	-2.60
9	0.1222	0.1198	0.1297	-1.96	6.14	-7.63
10	0.2141	0.2338	0.2159	9.20	0.84	8.29
11	0.2269	0.2249	0.2265	-0.88	-0.18	-0.71
12	0.0511	0.0511	0.0512	0.00	0.20	-0.20
13	0.3037	0.2995	0.3050	-1.38	0.43	-1.80
14	0.0070	0.0070	0.0070	0.00	0.00	0.00
15	0.2254	0.2222	0.2360	-1.42	4.70	-5.85
16	0.0716	0.0692	0.0714	-3.35	-0.28	-3.08
17	0.0214	0.0214	0.0214	0.00	0.00	0.00
18	0.1664	0.1846	0.2022	10.94	21.51	-8.70
19	0.2956	0.2888	0.2877	-2.30	-2.67	0.38
20	0.2379	0.2371	0.2388	-0.34	0.38	-0.71
21	0.0658	0.0625	0.0656	-5.02	-0.30	-4.73
22	0.0786	0.0798	0.0780	1.53	-0.76	2.31
23	0.1016	0.1231	0.1075	21.16	5.81	14.51
24	0.3110	0.3027	0.3158	-2.67	1.54	-4.15
25	0.2223	0.2187	0.2156	-1.62	-3.01	1.44
26	0.4371	0.4373	0.4321	0.05	-1.14	1.20

TABLE 4.6 (a). Average precision for each CISI query.

CISI						
	Average Precision			Percent Change in Average Precision (b - a) / a		
Query	(1) Single Term Indexing (ST)	(2) Phrase Indexing Non-syntactic (Nsyn.)	(3) Phrase Indexing Syntactic (Syn.)	(4) (a) ST vs. (b) Nsyn.	(5) (a) ST vs. (b) Syn.	(6) (a) Syn. vs. (b) Nsyn.
27	0.3333	0.3496	0.3447	4.89	3.42	1.42
28	0.2379	0.2273	0.2425	-4.46	1.93	-6.27
29	0.2003	0.1967	0.2106	-1.80	5.14	-6.60
30	0.4319	0.4307	0.4319	-0.28	0.00	-0.28
31	0.1476	0.1362	0.1366	-7.72	-7.45	-0.29
32	0.1763	0.1662	0.1697	-5.73	-3.74	-2.06
33	0.0489	0.0485	0.0528	-0.82	7.98	-8.14
34	0.1841	0.1925	0.1946	4.56	5.70	-1.08
35	0.2167	0.2221	0.2265	2.49	4.52	-1.94
37	0.1554	0.1849	0.1503	18.98	-3.28	23.02
39	0.0670	0.0643	0.0656	-4.03	-2.09	-1.98
41	0.3128	0.3116	0.2866	-0.38	-8.38	8.72
42	0.0952	0.0978	0.0941	2.73	-1.16	3.93
43	0.0406	0.0370	0.0398	-8.87	-1.97	-7.04
44	0.2825	0.2861	0.2833	1.27	0.28	0.99
45	0.1218	0.1220	0.1224	0.16	0.49	-0.33
46	0.3182	0.3180	0.3196	-0.06	0.44	-0.50
49	0.0977	0.0840	0.0921	-14.02	-5.73	-8.79
50	0.4811	0.4705	0.4707	-2.20	-2.16	-0.04
52	0.8339	0.7784	0.8297	-6.66	-0.50	-6.18
54	0.1258	0.1451	0.1399	15.34	11.21	3.72
55	0.9476	0.9268	0.9399	-2.19	-0.81	-1.39
56	0.1442	0.1399	0.1438	-2.98	-0.28	-2.71
57	0.1418	0.1427	0.1514	0.63	6.77	-5.75
58	0.5044	0.5293	0.5281	4.94	4.70	0.23

TABLE 4.6 (b). Average precision for each CISI query.

CISI						
Query	Average Precision			Percent Change in Average Precision (b - a) / a		
	(1) Single Term Indexing (ST)	(2) Phrase Indexing: Non-syntactic (Nsyn.)	(3) Phrase Indexing: Syntactic (Syn.)	(4) (a) ST vs. (b) Nsyn.	(5) (a) ST vs. (b) Syn.	(6) (a) Syn. vs. (b) Nsyn.
61	0.0463	0.0434	0.0420	-6.26	-9.29	3.33
62	0.6668	0.6634	0.6420	-0.51	-3.72	3.33
65	0.5559	0.5879	0.5684	5.76	2.25	3.43
66	0.5926	0.5755	0.5819	-2.89	-1.81	-1.10
67	0.1392	0.1572	0.1566	12.93	12.50	0.38
69	0.2206	0.2492	0.2547	12.96	15.46	-2.16
71	0.3393	0.3124	0.3021	-7.93	-10.96	3.41
76	0.5603	0.5952	0.5867	6.23	4.71	1.45
79	0.1869	0.1787	0.2077	-4.39	11.13	-13.96
81	0.0847	0.0783	0.0699	-7.56	**-17.47**	12.02
82	0.0704	0.0573	0.0640	**-18.61**	-9.09	-10.47
84	0.2094	0.1962	0.2024	-6.30	-3.34	-3.06
90	0.1528	0.1652	0.1639	8.12	7.26	0.79
92	0.0810	0.0798	0.0810	-1.48	0.00	-1.48
95	0.1415	0.1217	0.1415	-13.99	0.00	-13.99
96	0.3080	0.3839	0.4315	24.64	*40.10*	-11.03
97	0.5715	0.5421	0.5715	-5.14	0.00	-5.14
98	0.3865	0.3368	0.4120	-12.86	6.60	-18.25
99	0.3470	0.3078	0.3504	-11.30	0.98	-12.16
100	0.0244	0.0235	0.0248	-3.69	1.64	-5.24
101	0.5000	1.0000	0.5000	*100.00*	0.00	*100.00*
102	0.6016	0.5975	0.5977	-0.68	-0.65	-0.03
104	0.0661	0.0592	0.0634	-10.44	-4.08	-6.62
109	0.2469	0.3009	0.2651	21.87	7.37	13.50
111	0.7365	0.6721	0.7289	-8.74	-1.03	-7.79

TABLE 4.6 (c). Average precision for each CISI query.

Columns (1)-(3) contain average precision values for single term, non-syntactic, and syntactic phrase indexing, respectively. Columns (4)-(6) indicate the percent change in average precision.

This data shows that for both collections, the effects of both syntactic and non-syntactic phrase indexing are quite variable. For CACM, for example, the change in average precision due to non-syntactic phrase indexing in comparison to single term indexing ranges from a maximum increase of +474% for query 20, to a maximum decrease of -69% for query 48. For syntactic phrase indexing, the range is from +158% for query 8 to -36% for query 32. In each column, maximum increases are given in italics, and maximum decreases are given in boldface. A broad range of variation in performance is also exhibited by the CISI collection, but the range is not as extreme as that for CACM. The maximum increase in average precision for non-syntactic phrase indexing in comparison to single term indexing is +100% for query 101, whereas the maximum decrease is -18% for query 82. For syntactic phrase indexing in comparison to single term indexing, the range is from +40% for query 96 to -17% for query 81.

Table 4.7 summarizes the performance of individual queries further. Taking a change in average precision of 5% as a difference threshold, each query can be classified according to whether it performs better, equivalently, or worse for each pair of indexing methods. For example, the first row of part (a) of Table 4.7 compares syntactic phrase indexing to single term indexing and non-syntactic phrase indexing for the CACM collection. The first cell of this row shows that 23 queries perform better with syntactic phrase indexing than with single term indexing. That is, 23 queries have an increase in average precision of 5% or more with syntactic phrase indexing in comparison to single term indexing. In addition, 20 queries have the same level of performance, and 9 queries perform worse with syntactic phrase indexing in comparison to single term indexing.

Indexing Methods ↓ vs. →	Single Terms	Non-syntactic Phrases
Syntactic Phrases	23 better 20 same 9 worse	12 better 6 same 34 worse
Non-syntactic Phrases	39 better 4 same 9 worse	
	(a) CACM	
Syntactic Phrases	16 better 53 same 7 worse	22 better 47 same 7 worse
Non-syntactic Phrases	13 better 44 same 19 worse	
	(b) CISI	

TABLE 4.7. Summary of relative performance of three indexing methods. Difference threshold: 5.0% change in average precision.

The figures in Tables 4.1 and 4.7 are relatively consistent in their portrayal of the relative effectiveness of syntactic phrase indexing, non-syntactic phrase indexing, and single term indexing for the CACM collection. That is, both views of performance indicate that phrase indexing performs better than single term indexing, and that non-syntactic phrase indexing is more effective than syntactic phrase indexing.

The situation is somewhat different for the CISI collection, however. The average precision figures presented in Tables 4.1 and 4.2 indicate that the three indexing methods are essentially equivalent in retrieval effectiveness. There is a very slight indication that phrase indexing is an improvement over single term indexing, and a similarly slight indication that non-syntactic phrase indexing is more effective than syntactic indexing. The figures in part (b) of Table 4.7 present a similar picture in that phrase indexing appears to have only a slight influence on retrieval effectiveness. This is indicated by the substantial proportion of queries that perform equivalently under phrase indexing and single term indexing. However, the figures in Table 4.7 lead to quite different conclusions about the relative value of syntactic and non-syntactic phrase indexing. With syntactic phrase indexing, 16 queries perform better than with single term indexing, and only 7 perform worse. In contrast, with non-syntactic phrase indexing, 13 queries perform better than with single term indexing, and 19 perform worse. Further, with syntactic phrase indexing, 22 queries perform better than with non-syntactic phrase indexing, whereas only 7 queries perform worse with syntactic phrase indexing.

The information in part (b) of Table 4.7 thus provides some evidence that syntax-based phrase indexing may offer some advantages over non-syntactic phrase indexing.

4.1.3. Analysis of the Performance of some Representative Queries

In order to to provide further insights into how syntactic and non-syntactic phrases influence document ranking, this section compares the behavior of some representative queries from both collections. The queries were chosen in order to illustrate situations in which non-syntactic phrases perform better than syntactic phrases, as well as situations in which syntactic phrases perform better than non-syntactic phrases.

4.1.3.1. Non-syntactic phrases better than syntactic phrases

The queries examined in this section were chosen on the basis of the number of relevant documents retrieved at a rank of 30 or higher. These are queries for which the non-syntactic phrase indexing method succeeded in retrieving more relevant documents at high ranks than either single term indexing, or syntactic phrase indexing.

The text of Query 13 from the CISI collection appears in Figure 4.1. With regard to the number of relevant documents retrieved in the top 30, this query represents one of the largest contrasts in the performance of single term indexing and the two phrase indexing methods. With non-syntactic phrase indexing, this query retrieves 12 relevant documents in the top 30, What criteria have been developed for the objective evaluation of information retrieval and dissemination systems?

FIGURE 4.1. Text of CISI query 13.

Non-syntactic Phrases		**Syntactic Phrases**	
criteria	developed	objective	evaluation
developed	objective	system	evaluation
objective	evaluation	information	system
evaluation	information	retrieval	system
information	retrieval	dissemination	system
dissemination	retrieval	information	retrieval
dissemination	system	information	dissemination

FIGURE 4.2. Phrases identified in CISI query 13.

Query Number	Descriptor Number	Weight	Descriptor Type	Phrase Descriptor
		Non-syntactic Phrases		
13	10030	0.3837	1	dissem retrief
13	16894	0.2390	1	evalu inform
13	16949	0.3715	1	criter develop
13	33593	0.3353	1	dissem system
13	35689	0.3281	1	develop object
		Syntactic Phrases		
13	27795	0.3915	1	object evalu
13	36467	0.3353	1	dissem system

FIGURE 4.3. Non-syntactic and syntactic phrase subvectors for CISI query 13.

whereas syntactic phrase indexing retrieves only ten, and single term indexing retrieves just nine.

Figure 4.2 shows the phrases identified by the two phrase indexing methods in unstemmed form. After restrictions on the document frequency of phrases are applied, the final vectors contain the phrases shown in Figure 4.3.

The nine relevant documents retrieved in the top 30 with single term indexing are also retrieved in the top 30 by non-syntactic phrase indexing. In addition to these, non-syntactic phrase matches raise the ranks of relevant documents 134, and 137 due to matches on *evalu inform*. Relevant document 175 also moves into the top 30 due to a match on *dissem system*. Syntactic phrase indexing moves relevant documents 59 and 175 into the top 30 due to matches on *dissem system*. However, with syntactic phrase indexing, the rank of relevant document 474 is not maintained, so it moves out of the top 30.

Syntactic phrase indexing fails to perform as well as non-syntactic phrase indexing in this case because it fails to construct a phrase containing the stems *evalu* and *inform*. Absence of this phrase accounts for the failure of syntactic phrase indexing to move relevant documents 134 and 137 into the top 30, as well as its failure to maintain the rank of relevant document 474. Such a phrase could not be constructed from query 13 by the syntactic phrase indexing method without violating the basic strategy of constructing phrases only from words that are related as head and modifier.

Though for this query non-syntactic phrases retrieve more relevant documents at high ranks than syntactic phrases, the syntactic method moves relevant document 59 into the top 30, whereas the non-syntactic method does not. This is because the syntax-based method constructs the phrase descriptor *dissem system* from the text phrase in (4.1).

(4.1) Selective *Dissemination* of Information (SDI) *Systems*

In order for the non-syntactic method to identify this phrase, the value of the proximity parameter would have to be increased from one to three. As indicated in chapter 2, however, less restrictive proximity values have substantial negative effects on overall retrieval performance for CISI.

Query 21 from the CACM collection has 11 relevant documents. Of these, single term indexing does not retrieve any at a rank of 30 or higher. Non-syntactic phrase indexing, however retrieves four in the top 30, whereas syntactic phrase indexing retrieves just one in the top 30. Of all CACM queries, this one has the largest increase in the number of relevant documents retrieved in the top 30 by non-syntactic indexing in comparison to single term indexing and syntactic phrase indexing.

The text of this query appears in Figure 4.4. Because of the highly unrestrictive nature of the non-syntactic phrase construction method applied to the CACM collection, a large number of non-syntactic phrases are assigned. The final subvector containing non-syntactic phrases is shown in Figure 4.5.

Syntactic phrases appear in Figure 4.6. Although the three phrases *class reductions*, *complete reductions*, and *class complete* are identified by the syntactic phrase indexing rules, they are not assigned as phrase descriptors because they do not occur in the document collection.

computational complexity, intractability, class-complete reductions, algorithms and efficiency

FIGURE 4.4. Text of CACM query 21.

Query Number	Descriptor Number	Weight	Descriptor Type	Phrase Descriptor
21	4441	0.3272	1	clas comples
21	4456	0.3104	1	clas complec
21	4611	0.3286	1	complec comples
21	10141	0.2892	1	clas effici
21	10284	0.3908	1	algorithm intract
21	10924	0.3009	1	effici reduc
21	14518	0.4116	1	comput intract
21	14580	0.5023	1	complec intract
21	16394	0.5126	1	intract reduc
21	19419	0.2211	1	complec comput
21	20237	0.2002	1	algorithm complec
21	20253	0.2171	1	algorithm comples
21	20772	0.2196	1	clas comput
21	25816	0.2105	1	algorithm reduc
21	25945	0.3389	1	comples reduc
21	28039	0.3074	1	comples effici
21	29671	0.2313	1	comput reduc
21	30050	0.2906	1	complec effici
21	30329	0.3221	1	complec reduc
21	30753	0.1998	1	comput effici
21	32741	0.1988	1	algorithm clas
21	33355	0.1790	1	algorithm effici
21	38671	0.2379	1	comples comput
21	39034	0.3206	1	clas reduc

FIGURE 4.5. Non-syntactic phrase subvector for CACM query 21.

The non-syntactic phrases that are responsible for retrieving four relevant documents in the top 30 are listed in Figure 4.7. Of these 16 phrases, the syntactic method identifies four equivalent phrases. However, because the syntactic method fails to identify three of these in the documents, they are not assigned as phrase descriptors. This is a major part of the reason that syntactic phrases fail to retrieve as many documents at high ranks as non-syntactic phrases do.

It is instructive to examine the text of the documents containing the non-syntactic versions of these phrases in order to determine why the syntactic phrase construction procedure did not identify them. Relevant document 2701, which was retrieved at rank 2 with non-syntactic phrase indexing, contains the phrase descriptors *comples reduc* and *class reduc*.[3] The sources of these phrase descriptors appear in boldface in the text of this document in Figure 4.8. Though the elements of these phrase descriptors are not related in any meaningful way in the document, they help to improve the rank of this relevant document. A similar situation arises in document 2703, where a phrase is constructed from the word *computationally*, occurring in the abstract, and *complexity*, occurring in the title. The text of this document appears in Figure 4.9.

[3] The stemmed forms *comples*, *reduc*, and *class* correspond to unstemmed forms *complete*, *reducible*, and *class(es)*, respectively.

Syntactic phrases identified	
computational	complexity
class	reductions
complete	reductions
class	complete

Syntactic phrase assigned as a descriptor	
comput	complec

FIGURE 4.6. Syntactic phrases in CACM query 21.

Phrase Descriptor		Text Source	
algorithms	class	algorithms	class
algorithms	complec	algorithms	complexity
algorithms	comples	algorithms	complete
algorithms	effici	algorithms	efficiency
algorithms	intract	algorithms	intractibility
algorithms	reduc	algorithms	reductions
class	complec	class	complexity
class	reduc	class	reductions
complec	comples	complexity	complete
complec	comput	complexity	computational
complec	intract	complexity	intractibility
comples	reduc	complete	reductions
comput	effici	computational	efficiency
comput	intract	computational	intractibility
comput	reduc	computational	reductions
effici	reduc	efficiency	reductions

FIGURE 4.7. Non-syntactic phrase descriptors from CACM query 21 that match relevant documents retrieved at rank 30 or higher.

Title:
A Fast and Usually Linear Algorithm for Global Flow Analysis (Abstract only--**Complete** paper JACM 23,1 January, 1976)

Abstract:
A new algorithm for global flow analysis on **reducible** graphs is presented. The algorithm is shown to treat a very general **class** of function spaces. For a graph of e edges, the algorithm has a worst case time bound of O(e log e) function operations. It is also shown that in programming terms, the number of operations is proportional to e plus the number of exits from program loops. Consequently a restriction to one-entry one-exit control structures linearity. The algorithm can be extended to yet larger **classes** of function spaces and graphs by relaxing the time bound. Examples are given of code improvement problems which can be solved using the algorithm.

FIGURE 4.8. Text of CACM document 2701.

Title:
The Intrinsically Exponential **complexity** of the Circularity Problem for Attribute Grammars

Abstract:
Attribute grammars are an extension of context-free grammars devised by Knuth as a mechanism for including the semantics of a context-free language with the syntax of the language. The circularity problem for a grammar is to determine whether the semantics for all possible sentences (programs) in fact will be well defined. It is proved that this problem is, in general, **computationally** intractable. Specifically, it is shown that any deterministic algorithm which solves the problem must for infinitely many cases use an exponential amount of time.An improved version of Knuth's circularity testing algorithm is also given, which actually solves the problem within exponential time.

FIGURE 4.9. Text of CACM document 2703.

With queries of this kind, it appears that the major strength of non-syntactic phrase indexing is its unrestrictive quality. That is, from a query that is short and well-focused, practically any pair of words yields a reasonable phrase, regardless of proximity or syntactic relationship. Examples from this query include *computational efficiency*, *algorithms complexity*, and *computational intractability*. Phrases like these are not identified by the syntax-based phrase construction rules because the elements do not enter into a relationship of modification with one another.

Though the less restrictive nature of the non-syntactic approach does yield phrase descriptors that enhance retrieval effectiveness in some cases, it also has some negative effects. For example, relevant document 2932 has a retrieval rank of 72 with single term indexing. With syntactic phrase indexing, a match on the phrase descriptor *comput complec* raises this document to a rank of 20. The source of this phrase is the title of the document, *Complexity of Computations* (see Figure 4.10). With non-syntactic phrase indexing, however, the rank of this document is lowered to 84, even though it matches the corresponding non-syntactic query phrase, *complec comput*. The problem here is that many of the phrase descriptors that result from unrestricted combining of word pairs, as is done with the non-syntactic phrase construction method, frequently match on phrases resulting from superfluous combinations of words in non-relevant documents. The result is that many non-relevant documents get retrieved at higher ranks.

Title:
Complexity of Computations

Abstract:
The framework for research in the theory of complexity of computations is described, emphasizing the in terrelation between seemingly diverse problems and methods. Illustrative examples of practical and theoretical significance are given. Directions for new research are discussed.

FIGURE 4.10. Text of CACM document 2932.

Title:
The Self-Judgment Method of Curve Fitting

Abstract:
A **computer**-oriented method for processing and communicating numerical data is described. The Instrument Reliability Factors (IRF), which exactly define the limits of reliability of each measured item of information, are used to **compute** the Maximum Permitted Error (MPE) associated with each values of each ordinate. The Self-Judgment Principle (SJP) is used to discard wrong information and to **compute** mean values of the parameters and their MPE's in terms of the IRF. Data compatibility tests with any number of different equations can be made quickly. Otherwise **intractable** problems are easily solved, and the design of many experiments is greatly simplified. The **computational** and mathematical techniques used to **reduce** bias in the SJP are discussed. Inadequacies in the statistical and graphical methods of curve fitting are noted.

FIGURE 4.11. Text of CACM document 1206.

An example of this is non-relevant document 1206. With single term indexing and syntactic phrase indexing, this document was not retrieved in the top 30. With non-syntactic phrase indexing, however, it was retrieved with a rank of nine due to fortuitous matches on *comput intract*, *intract reduc*, and *comput reduc*. From the text of document 1206 (see Figure 4.11), it is clear that these phrase descriptors are not constructed from meaningful combinations of words, and they are not good indicators of document content.

By incorrectly increasing the ranks of several non-relevant documents such as this one, the non-syntactic phrase indexing method causes a clearly relevant document (2932) to be lowered in rank.

4.1.3.2. Syntactic phrases better than non-syntactic phrases

For the queries examined in this section, the syntactic phrase indexing method retrieves more relevant documents at a rank of 30 or higher than either single term indexing or non-syntactic phrase indexing.

With only single term descriptors, CACM query 48 retrieves two of its 12 relevant documents in the top 30, document 2325 at rank 6, and document 1797 at rank 8. With the addition of non-syntactic phrases, only one relevant document has a rank of 30 or higher, whereas syntactic phrase indexing retrieves five documents in the top 30. The text of query 48 is given in Figure 4.12, and the syntactic phrase descriptors appear in Figure 4.13. The non-syntactic phrase indexing procedure assigns a total of 95 phrase descriptors to this query. All of the syntactic phrases are included in this set with the exception of *algorithm gener*, from *generating ... algorithms*, and *lin program*, from *linear programming*. These two phrases are eliminated from the set of non-syntactic phrases because their document frequencies exceed the threshold of 90.

The use of computer science principles (e.g. data structures, numerical methods) in generating optimization (e.g. linear programming) algorithms. This includes issues of the Khachian (Russian, ellipsoidal) algorithm and complexity of such algorithms.

FIGURE 4.12. Text of CACM query 48.

algorithm	gener
optim	algorithm
comput	sci
lin	program
numer	method
algorithm	complec

FIGURE 4.13. Syntactic phrases in CACM query 48.

The three additional relevant documents retrieved in the top 30 with syntactic phrase indexing are all due to matches on the phrase *lin program*. Document 1797, which had a rank of 8 with single term indexing, also matches on this phrase, raising it to a rank of 4. Using the syntax-based phrase indexing method, *lin program* has a document frequency of 33, so it is not eliminated on the basis of its document frequency. The corresponding non-syntactic phrase has a document frequency of 98. A phrase match on *comput sci* maintains the rank of 6 for document 2325. With non-syntactic phrase indexing, document 2325 rises to rank 2 as a result of 12 phrase matches. Document 1797, however, descends to rank 110.

It might appear that the absence of the phrase *lin program* from the set of non-syntactic phrase descriptors would account for the substantial drop in rank of document 1797, and the failure of the non-syntactic phrases to retrieve the additional three relevant documents that were retrieved due to matches on the syntactic phrase descriptor *lin program*. However, the results of an additional retrieval experiment provide evidence that the absence of *lin program* is not the primary cause of the poor performance of non-syntactic phrase indexing on this query.

If the non-syntactic phrase indexing procedure is applied to the document and query collections without using any restrictions on the document frequency of phrases, 113 phrase descriptors are assigned to query 48, and the phrase *lin program* is among them. When this phrase indexing method is used, relevant documents 2325 and 1797 are retrieved in the top 30, just as with single term indexing. Due to matches on *lin program* and two other phrases, document 1797 is retrieved at rank 22. However, even with *lin program* as a phrase descriptor, the additional three relevant documents retrieved in the top 30 by syntactic phrases are not retrieved in the top 30.

The poor performance of the non-syntactic phrase indexing method on this query is due primarily to the method's lack of selectivity in assigning phrase descriptors. An excessive number of phrase descriptors are assigned, and a significant proportion of them are not good indicators of the content of either documents or queries. Non-relevant document 3200 illustrates the effect that phrases of low quality can have on retrieval performance. With single term indexing, this document is not retrieved at a rank of 30 or above. With non-syntactic phrase indexing, however, 15 of its 29 phrase descriptors match phrase descriptors of query 48, so it is retrieved with a rank of 19. The matching phrase descriptors appear in Figure 4.14; the sources of the elements of these phrases are in boldface in the text of the document in Figure 4.15. Many of these phrase descriptors do not appear to be good indicators of the content of either the document or query.

comput	lin	inclus	numer
comput	numer	inclus	optim
comput	optim	lin	numer
gener	inclus	lin	optim
gener	lin	numer	optim
gener	numer	numer	program
gener	optim	optim	program
inclus	lin		

FIGURE 4.14. Phrases in CACM document 3200.

Title:
A FORMAC **Program** for the Solution of **Linear** Boundary and Initial Value Problems

Abstract:
A **computer program** is described which has been developed for obtaining approximate solutions to **linear** initial and boundary-value problems involving differential equations. For each problem, input to the **program includes**: 1. The equations (in symbolic form) to be satisfied - the differential equations, equations describing auxiliary conditions such as boundary conditions, etc. 2. A **numerical** description of the regions in which each of the equations are to be satisfied. 3. Sets of functions (in symbolic form) to be used in **linear** combinations to approximate the solution functions. Give the above input, the **program generates** an approximation to the solutions of the specified problemm in terms of the specified functions which is **optimum** in the least-squares sense.

FIGURE 4.15. Text of CACM document 3200.

The drop in rank of relevant document 1797 from 8 with single term indexing to 110 with non-syntactic phrase indexing, and the failure of non-syntactic phrases to retrieve the additional three relevant documents retrieved in the top 30 with syntactic phrases is thus not caused primarily by the lack of the single, semantically appropriate phrase descriptor *lin program*. Rather, these shortcomings are the result of many matches between superfluous phrase descriptors in the query and many non-relevant documents like document 3200.

Query 11 from the CISI collection has 127 relevant documents. Single term indexing retrieves nine of these at ranks of 30 or higher. Syntactic phrases increase this count to ten, while non-syntactic phrases reduce it to eight. The eight documents retrieved in the top 30 using non-syntactic phrases were also retrieved in the top 30 using single terms alone. However, with non-syntactic phrases, relevant document 166 was displaced from the top 30 due to phrase matches that moved six non-relevant documents into the top 30 that did not achieve such high ranks with single term indexing. All of the relevant documents retrieved in the top 30 with single term indexing were also retrieved with syntactic phrases. In addition, another relevant document moved into the top 30 due to a phrase match. Syntactic phrases did not match on any non-relevant documents retrieved in the top 30.

The text of query 11 is given in Figure 4.16. The phrases identified by the syntactic and non-syntactic phrase indexing procedures appear in Figure 4.17. The actual query subvectors are given in Figure 4.18. Phrases in Figure 4.17 that are not assigned as phrase descriptors have been eliminated

What is the need for information consolidation, evaluation, and retrieval in scientific research?

FIGURE 4.16. Text of CISI query 11.

Syntactic Phrases		**Non-syntactic Phrases**	
consolidation	need	information	need
evaluation	need	consolidation	information
retrieval	need	consolidation	evaluation
information	consolidation	evaluation	retrieval
information	evaluation	retrieval	scientific
information	retrieval	research	scientific
scientific	research		
research	consolidation		
research	evaluation		
research	retrieval		

FIGURE 4.17. Phrases in CISI query 11.

Query Number	Descriptor Number	Weight	Descriptor Type	Phrase Descriptor
		Non-syntactic Phrases		
11	6020	0.2451	1	evalu retrief
11	26183	0.1846	1	retrief sci
		Syntactic Phrases		
11	22836	0.1969	1	research retrief
11	26416	0.1924	1	inform evalu
11	35114	0.2210	1	retrief need
11	35775	0.2319	1	research evalu

FIGURE 4.18. Non-syntactic and syntactic phrase subvectors for CISI query 11.

either because of document frequency restrictions, or because they do not occur in any of the documents.

The additional relevant document retrieved with syntactic phrase indexing (document 1098) is due to a match on *inform evalu*. The syntactic phrase indexing rules are able to construct this phrase because of the strategies of (a) associating heads with modifiers, and (b) distributing the premodifier of the first conjunct of a group of conjoined noun phrases over all the conjuncts of the noun phrase (see sections 3.2.1 and 3.4.2.3). The non-syntactic phrase indexing procedure does not identify this phrase in either the query or in relevant document 1098 because of the restrictive proximity requirements used for the CISI collection.

The non-syntactic phrase indexing method resulted in moving six non-relevant documents into the top 30 ranks due to matches on two phrases, *evalu retrief* and *retrief sci*. The sources of these descriptors are displayed in (4.2).

(4.2)	61:	it **evaluates retrieval** performance relative to random searching
	509:	the **evaluation of retrieval** strategies
	523:	the design, operation and **evaluation of retrieval** systems
	525:	establishing, operating, and **evaluating retrieval** systems
	634:	the best way to **evaluate a retrieval** system for **evaluating retrieval** systems on this basis
	686:	publication, distribution, storage, and **retrieval of scientific** information

Though *retrief sci* in document 686 is not a good indicator of document content, each of the occurrences of the phrase *evalu retrief* identified in these documents is a semantically appropriate indicator of the content of the document. However, this phrase is not appropriate as an indicator of the content of the query. The query is concerned more with the evaluation of information, or the evaluation of the quality of various sources of information, rather than with evaluation of information retrieval systems. This is indicated by the text of the query itself, as well as the documents that are judged to be relevant. As an example, the title and abstract of relevant document 1098 (see Figure 4.19) indicate that the study focuses on evaluation of information sources (e.g., periodicals and bibliographies) rather than information retrieval systems.[4]

The essence of the problem illustrated by this example is that the non-syntactic phrase indexing method has constructed identical phrase descriptors from textual sources that differ significantly in meaning. This is an example of incorrect normalization. The resulting phrase matches move non-relevant documents to higher retrieval ranks. Because the syntactic phrase indexing method makes use of information about the syntactic structure of text, it is

[4] It should be noted that even though *evalu inform* is a semantically appropriate descriptor for the first five document appearing in (4.2), the syntactic phrase indexing rules do not succeed in identifying them. This is due to difficulties related to analyzing the structure of complex noun phrases, as discussed in section 3.4.6.1, and the currently limited treatment of verbal constructions. However, with appropriate extensions of the strategy of raising heads to modifiers (see 3.4.6.1), and more adequate treatment of verbal constructions, it should be possible to solve most of these problems. Recognition of this phrase in these documents would not, however, change the degree of similarity between query 11 and these documents, since the phrase is correctly not assigned as a descriptor to the query.

able to recognize that even though *evaluation* and *retrieval* are in close proximity in the query, they are not related syntactically as head and modifier, and therefore that they should not be combined to form a phrase descriptor. On the other hand, the syntactic structure of the query indicates that *information* and *evaluation* are related as modifier and head, and therefore that they should be combined to form a phrase.

Title:
Concerning the Criterion for **Evaluation** of Current Secondary **Information**

Abstract:
The findings are described of a study aimed at determining the prospects and methods for improving the system of current bibliographic information.. The analysis has shown that the existing criteria for evaluation of special bibliographies (scope, coverage, arrangement, speed of announcement, etc.) are inadequate for an unbiased characterization of their exhaustivity and subject contents.. This hampers a correct choice of the sources of secondary information and leads to duplication, parallelisms and loss of information.. Judgements of the leading Soviet and foreign bibliographers relating to the problems under consideration are reviewed, which are all essentially in favor of a reconstruction of the publishing processes, issuing of scientific publications on a world scale, and algorithmization of the information processes.. It is suggested that the first objective of research should be a method of comparative evaluation of periodicals..

FIGURE 4.19. Text of CISI document 1098.

4.2. Other Phrase Indexing Experiments

In order to place this study into a broader context, this section compares the syntactic and non-syntactic phrase indexing procedures presented in chapters 2 and 3 with several previous studies. Though a substantial number of methods have been proposed for constructing phrases for use as content indicators (see sections 1.3.2 and 1.4.1), few of these proposals have been accompanied by experiments that test the influence of phrase descriptors on retrieval effectiveness. This section discusses only work that has involved substantive retrieval experiments.

Three categories of phrase construction methods can be recognized: (a) non-syntactic methods, (b) methods involving simplified syntactic processing, and (c) methods involving actual parsing of document and/or query text. For purposes of this discussion, non-syntactic and syntactic phrase indexing can be distinguished as follows:

(a) Non-syntactic phrase indexing takes into consideration only the statistical and positional characteristics of words in text. Such characteristics may include the frequency of a word in a document or in a collection, or the cooccurrence characteristics (including proximity) of two or more terms. A phrase dictionary constructed by any available means may also be used.

(b) Syntactic phrase indexing involves at least the use of information about the grammatical categories of words and their patterns of cooccurrence. Any information used in non-syntactic phrase indexing may also be used. Methods involving manual identification of syntactically correct natural language phrases are also included in this category.

4.2.1. Non-syntactic Methods

The experiments conducted by the SMART project in the mid-1960s are among the earliest tests of the use of phrase descriptors.[5] The approach made use of a dictionary of phrases that was either compiled manually by a subject

---

[5]A general overview of the non-syntactic phrase indexing procedures used by the SMART project at this time can be found in Salton and Lesk (1965), Salton (1968:23-24, 33-38, 47-49, 93-96, 334-340), and Salton and Lesk (1971:117, 126-127). Details of the algorithms used for phrase recognition are available in Lesk and Evslin (1964), Evslin (1965:3-4), and Shapiro (1965).

expert, or constructed on the basis of statistical term associations. A phrase was assigned as a descriptor if all of its elements were found to cooccur in a document. Most experiments required that the terms cooccur in the same sentence of a document, but it appears that in some cases, simple cooccurrence in the document was held to be sufficient. It was not required that the terms be adjacent or in a specified order in the text.[6] Using three different document collections, experiments were conducted to evaluate the effectiveness of this phrase indexing procedure. Using both the phrase dictionary and term associations, there was no consistent evidence of improvement offered by this method of phrase indexing. Small improvements in precision were achieved at certain levels of recall, but for only one collection was the improvement statistically significant for the term association method, and none of the improvements resulting from the phrase dictionary were significant (Salton and Lesk 1968:24, 26-27, 30-31).

More recent work by Salton and his co-workers has yielded experimental results that are among the best reported to date for experiments on automatic phrase indexing. A number of experiments based on the term discrimination model were conducted, and the details of the phrase identification procedure differ somewhat in each study (Salton, Yang, and Yu 1974; Salton and Wong

---

[6] In the literature, a phrase recognized on this basis is called a *statistical phrase*. Note, however, that this name is not necessarily indicative of the method of phrase recognition (Salton 1968:93): "The term *statistical phrase* is thus used not because any statistical techniques are included in the phrase detection process, but by opposition to *syntactic phrase* where a definite syntactic relationship is assumed among the phrase components."

1976). However, the best overall retrieval results are those reported by Salton, Yang, and Yu (1975). This approach is the basis for the non-syntactic phrase indexing method presented in chapter 2.

For experimental purposes Salton, Yang, and Yu (1975) used as phrase descriptors only phrases occurring in the query collection. The criteria for identifying pairs of terms as phrases were: (1) terms must cooccur in a query or document and be separated by at most one other term, (2) at least one of the terms must be a high document frequency, poor discriminator, and (3) the elements of a phrase may not be identical. The phrase descriptor then replaced the two single term descriptors.

A summary of Salton, Yang, and Yu's experimental results is given in Table 4.8. This table contains two sets of comparisons for three small document collections. The row labeled "tf" compares the average precision attained with simple single term indexing with term frequency (tf) weights to the results of phrase indexing. This data appears in Salton, Yang, and Yu (1975), and it shows that phrase indexing yields an increase in average precision of between 17% and 39% over single term indexing with term frequency weights. The row labeled "tf×idf" compares the same phrase indexing results to results of single term indexing with weights calculated as a product of term frequency and inverse document frequency. These figures are based on results reported by Salton and Yang (1973). In comparing phrase indexing with single term indexing and this better weighting method, their phrase indexing still shows an increase in average precision, but the magnitude of the increase is much less. Rather than ranging between 17% and 39%, the range is from 6% to 20%. In comparing Salton, Yang, and Yu's results with the results of the current study, the tf×idf figures are more appropriate, since single term indexing with tf×idf weighting is used as the basic of reference for evaluating retrieval results.

Single Term Weighting Method	Average Precision					
	CRANFIELD 424 documents		MEDLARS 450 documents		TIME 425 documents	
	ST	PH	ST	PH	ST	PH
		0.4287		0.5468		0.6783
tf	0.3207	+32%	0.4158	+39%	0.5794	+17%
tf×idf	0.3788	+11%	0.4722	+20%	0.6440	+6%

TABLE 4.8. Average precision figures for single term indexing (ST) with tf and tf×idf weights and for Salton, Yang, and Yu's phrase indexing (PH). Percentages indicate changes in average precision attained by phrase indexing.

By comparing these results with the figures in Table 2.2 (section 2.3) it can be seen that Salton, Yang, and Yu's results are approximately equal to, or better than, the non-syntactic phrase indexing results obtained in this study. In the present study, the best average precision increase using non-syntactic phrases was 22.7% for CACM. This is comparable to Salton, Yang, and Yu's result for the MEDLARS collection. The average precision increases obtained for INSPEC and CRAN are close to the 11% increase obtained by Salton, Yang, and Yu for the CRANFIELD collection and their 6% increase for the TIME collection. The results obtained in this study for CISI and MED, however, are lower than any of the results obtained by Salton, Yang, and Yu.

In comparison to the syntax-based method used in this study, Salton, Yang, and Yu's results for the MEDLARS collection are substantially better than the 8.7% increase using syntactic phrases with the CACM collection. This level of improvement for CACM is, however, comparable to Salton, Yang, And Yu's result for the CRANFIELD and TIME collections. Like the non-syntactic results for the CISI collection, the increase in average precision achieved with syntactic phrases on the CISI collection are lower than any of the results obtained by Salton, Yang, and Yu.

4.2.2. Simplified Syntactic Methods

Martin Dillon (Dillon and Gray 1983, Dillon and McDonald 1983) has developed a phrase indexing system (FASIT) that makes use of syntactic information in a simplified way. The basic strategy is to determine the syntactic category of each text word by dictionary look-up, and then to match sequences of category symbols against a dictionary of acceptable patterns. Sequences of text words that match one of the patterns in the dictionary are used as phrase descriptors. The system also includes simple methods of phrase normalization and grouping of phrases that are likely to be related in meaning.

FASIT's effect on retrieval performance was evaluated by an experiment comparing it with two other indexing methods: (1) a simple automatic indexing procedure that used single term descriptors, a small stopword list, and consolidation of singular and plural forms, and (2) indexing based on a manually constructed thesaurus. The indexing procedures were applied to a collection of 250 documents on topics in library science and 22 natural language queries. Inverse document frequency (idf) weights and the cosine similarity function were used for retrieval with all three indexing methods. The results showed that both FASIT and single term indexing are better than the thesaurus indexing method, and that FASIT yields slightly better precision than single term indexing at most recall levels below 80%. The phrase indexing procedure proved to be slightly better than the single term procedure as indicated by increases in precision at most recall levels below 80%. At 40-60% recall, the increase in precision ranged between 3% and 7%.

It is difficult to directly compare these results with those of the present study, since different collections and weighting were used. However, when applied to the CACM, CRAN, and INSPEC collections, the non-syntactic procedure tested in this study appears to be as good as or better than Dillon's syntax-based procedure. Compared to the syntax-based phrase indexing results reported in chapter 3, Dillon's results are comparable to the improvement achieved for the CACM collection, and better than the results for CISI.

A precursor to Dillon's work is the system developed by Klingbiel (1973a, 1973b). The basic strategy is word class assignment and matching against a dictionary of stored patterns. Klingbiel's system is significantly less sophisticated than Dillon's in three respects: (1) assignment of word classes is less accurate since each word can belong to only one class, (2) no phrase normalization is done, and (3) no grouping of semantically related phrases is attempted.

Klingbiel and Rinker (1976) reported on some retrieval experiments that compared Klingbiel's automatic indexing method and manual indexing. The results showed that manual indexing yielded slightly better recall, and automatic indexing yielded slightly better precision. The differences were in the range of 4-7%. Since no experiments were done to compare Klingbiel's system to any other automatic indexing systems, Klingbiel's results cannot be compared to the results of the present study.

Croft (1986a) has tested a method of incorporating information about phrases that occur in natural language queries into a probabilistic retrieval model. Though Croft presents his approach as a method of incorporating information about term dependencies, rather than phrases, into the retrieval process, his approach is, in effect, equivalent to phrase indexing. This is also the case for Smeaton's work (see below).

For queries, phrases were generated manually from the query text, so these phrases correspond to syntactically correct natural language phrases.

In indexing the documents, phrase descriptors were not explicitly assigned. Rather, at retrieval time, simple cooccurrence of a set of dependent terms (that is, the elements of a query phrase) was taken to be indicative of the presence of the phrase. For example, the natural language phrase *hidden line* would be interpreted as indicating an important dependency relationship between the terms *hidden* and *line*, and thus the presence of both terms anywhere in a document would result in an increase in its retrieval rank.

Croft used single term indexing with idf weights as an approximation for a document ranking produced by a probabilistic retrieval model assuming term independence. This yielded an average precision of 0.2110. After correcting the idf ranking by taking into consideration the dependency relationships of elements of query phrases, the average precision rose to 0.2270, an increase of 7.6%.[7]

The 7.6% increase in average precision achieved by Croft's use of term dependencies derived manually from natural language query phrases is substantially lower than the 22.7% increase achieved by the non-syntactic phrase indexing procedure as applied to the CACM collection. Croft's 7.6% increase is closer to the 8.7% increase resulting from the syntactic phrase indexing method used in this study. In comparing Crofts results with those of this study, however, it should be pointed out that Croft compared his phrase-based

[7] These figures are based on data in Croft (1986a:75, Table 2). Average precision was calculated at recall levels 0.10-0.90.

results to single term indexing with simple idf weights, while the results of the current study are based on a comparison of phrase indexing and single term indexing using tf×idf weights. The use of idf vs. tf×idf weights is discussed further below.

Smeaton (1986) has conducted an experiment on the CACM collection with an indexing method that uses information about the syntactic structure of natural language query phrases to identify term dependencies. The general strategy for incorporating term dependency information into the retrieval process is essentially the same as Croft's method. First, natural language query phrases are identified by performing a manual syntactic analysis of the text of queries. Then, dependent pairs and triples of terms are identified in the natural language query phrases. For a particular query, dependencies are then identified in all relevant and non-relevant documents retrieved at ranks of 20 or higher in a single term retrieval run. Identification of dependent terms in documents is based on a much shallower syntactic analysis. The only requirement is that the elements of a set of dependent terms cooccur in a sentence, clause, or phrase in the document text. It is apparently not required that terms be related syntactically in the document text. The presence of a set of dependent terms in a document increases the retrieval rank of that document.

Smeaton's best term dependency results show an increase of 23.9% in average precision over single term indexing.[8] In comparing this result with the results of the present study, it is important to note that Smeaton (like Croft) compares the term dependency results to single term indexing with simple idf weights, rather than tf×idf weights, as has been done in this study. Table 4.9 compares retrieval results for the CACM collection for single term indexing using two different weighting methods, idf and tf×idf. Two different query collections were tested: (1) the 25 queries used by Smeaton,[9] and (2) the complete collection of 52 queries. Smeaton's 25 queries are a subset of the larger collection. This table shows that single term indexing with tf×idf weights performs better than single term indexing with idf weights for both query collections. The tf×idf weights yield an increase of 7.3% in average precision for the collection of 25 queries, and an increase of 21.0% for the collection of 52 queries. Using single term indexing with tf×idf weights as a point of comparison thus provides a more stringent basis for evaluating the performance of a phrase indexing method.

As can be seen from Table 4.10, the non-syntactic phrase indexing procedure yields an increase in average precision of 38.3% over single term indexing with tf×idf weights when applied to Smeaton's set of 25 queries. This 38.3% increase is higher than the 23.9% increase achieved by Smeaton's

---

[8] This calculation is based on Smeaton's data for single term indexing and his corrected results that yield the best average precision figures. The data is given in Smeaton (1986:107), Table 8, columns labeled "IDF Uncorrected" and "Corrected 5,10,5,10." Averages were calculated at recall levels 0.10-0.90.

[9] I am grateful to Alan Smeaton for providing me with information about the query col-

syntax-based procedure. The syntax-based procedure of chapter 3 yields an increase of 14.7% over single term indexing. Since the point of comparison here is single term indexing with tf×idf weights, this 14.7% increase may be roughly comparable to Smeaton's 23.9% increase over single term indexing with idf weights.

Query Collection	Single Term (idf)	Single Term (tf×idf)
Smeaton's 25	0.2079	0.2230 +7.3%
Standard 52	0.2153	0.2604 +21.0%

TABLE 4.9. Average precision for the CACM collection using single term indexing with two weighting methods and two query collections.

Query Collection	Single Term (tf×idf)	Non-syntactic Phrasing	Syntactic Phrasing
Smeaton's 25	0.2230	0.3083 +38.3%	0.2557 +14.7%
Standard 52	0.2604	0.3195 +22.7%	0.2830 +8.7

TABLE 4.10. Average precision for the CACM collection using two query collections. Single term indexing with tf×idf weights compared to non-syntactic and syntactic phrase indexing.

lection used for his experiments.

4.2.3. Syntactic Methods

In their review of the application of linguistics to information science, Sparck Jones and Kay characterize Salton's work on syntax-based content analysis as a major effort in the field (Sparck Jones and Kay 1973:105-106). This work is also distinctive in that some comparative evaluation of the content analysis method was done.

The objective of incorporating syntactic analysis into the process of content analysis was to recognize phrases consisting of pairs of descriptors that stand in specified relationships to one another (Salton 1966, Salton 1968:166-178).[10] The method also provided for a significant degree of both syntactic and lexical normalization. The procedure can be summarized as follows:

(1) Manually construct a thesaurus in which each class contains a set of related word stems.

(2) Manually construct a dictionary of phrases; these are called *criterion phrases* or *criterion trees*. Each entry in the criterion tree dictionary specifies a pair of descriptors (thesaurus classes), and a set of possible syntactic relationships that must hold between the pair of descriptors. The syntactic relationships could be one or more of: (a) head-modifier in a noun phrase, (b) subject-verb, (c) verb-object, or (d) subject-object.

(3) Perform a syntactic analysis of a query or document sentence (Kuno and Oettinger 1962).

---

[10] A related discussion of graphical representations of term relationships can be found in Salton (1962).

(4) Using an automatic structure matching procedure, match phrases in the sentence against phrases in the criterion tree dictionary. Assign a phrase as a descriptor if the document contains terms from the thesaurus classes specified by a criterion tree, and the syntactic relationship of the terms in the document matches one of the syntactic relationships specified in the criterion tree.

This procedure accomplishes significant lexical normalization, since each criterion tree entry specifies thesaurus classes rather than individual word stems. Syntactic normalization is accomplished because a single entry may allow more than one syntactic relationship to hold between the specified descriptors. For example, the three constructions *text analysis*, *analysis of text*, and *analyzes text* could all be mapped to the same criterion tree entry.

The syntactic phrase indexing procedure was evaluated by comparing its retrieval performance with that of a non-syntactic phrase indexing method. Based on a retrieval experiment involving 17 queries, the syntactic method resulted in slightly lower precision than the non-syntactic method at all levels of recall (Salton 1968:198). The change in average precision was about 7%.

Possible explanations for the failure of the syntactic procedure to realize more substantial improvements are of two types. One is that the syntactic criteria are excessively stringent, and therefore prohibit the assignment of phrase descriptors when they are actually quite appropriate. This is the point of view taken by Salton (1968:198). In contrast, Sparck Jones and Kay (1973:106) express quite strongly the opinion that the use of a phrase dictionary limits the number of possible phrases so severely that significant improvement cannot be achieved. It should also be pointed out that the small scale of the experiment (17 queries) does not provide a firm basis for drawing final conclusions concerning the possible usefulness of syntax in content analysis.

A more recent effort to construct multi-term descriptors on a syntactic basis is that of Tait and Sparck Jones (Tait 1984; Sparck Jones and Tait 1984a, 1984b). Their work goes beyond the use of strictly syntactic information, however, since it incorporates general, non-domain specific information provided by the parser developed by Boguraev (1979). Their objective is to identify syntactically related groups of terms in query texts, and then to generate a set of variant phrases that express essentially the same concept. Each of these variants can then be used as a search term, and matched against the text of documents. By identifying phrase descriptors, the system provides high precision query terms. By generating variant forms of the phrase descriptors, decreases in retrieval performance due to losses in recall can be minimized.

Their analysis procedure makes it possible to identify source phrases like *circuit details*, and *retrieval of information*. The variant generation procedure can then produce related phrases like *details about circuits*, *details of a circuit*, and *information retrieval*.

In order to successfully generate the noun phrases containing prepositional phrase modifiers from the corresponding noun phrase with nominal premodifier, it seems clear that some degree of semantic information must be provided in order to select the appropriate prepositions. However, some of their apparently more complex examples of variant generation could be accomplished in a straightforward way from a purely syntactic surface structure parse. This is the case, for example, with the phrase *high frequency oscillator using slow switching germanium transistors* (Tait 1984). A surface structure parse yields sufficient information to construct the phrase *oscillator using transistors*, and to avoid constructing the incorrect *high frequency transistors*. Examples such as this inevitably lead one to ask whether the level of general semantic information currently available to this system actually provides significant capabilities beyond that provided by a simpler syntactic parse.

Another question prompted by this approach is whether or not the strategy of query term variant generation and text searching is actually preferable to indexing of both documents and queries using a process that identifies phrases in document and query texts and normalizes them to descriptors of a standard form. There is certainly some validity to Sparck Jones and Tait's argument that linguistically sophisticated analysis of large document collections is currently not practical. There is, however, reason to believe that the variant generation approach may not be viable.

The success of the variant generation approach depends on correctly generating a large proportion of the most likely paraphrases of each source expression found in a query. Relatively simple paraphrases of the kind exhibited in the papers by Sparck Jones and Tait are certainly useful and appropriate. However, to be fully successful, a great variety of other kinds of related expressions would need to generated, and it may not be possible to accomplish this. Sparck Jones and Tait (1984a:63) provide an example that illustrates a problem of this kind. From the query phrase *retrieval of information*, it would be difficult to generate all useful related phrases like *retrieval of relevant information.* Constructions involving conjunctions pose a related problem. For example, a query phrase like *preparation of extracts* does not provide an adequate basis for generating a phrase that would match *preparation and evaluation of computer-prepared abstracts and extracts.* Examples of this kind are not uncommon. For example, a sample of 20 out of 1460 documents in the CISI collection contains 25 such expressions.

The variant generation approach certainly does have some advantages. However, the difficulties related to successful generation of a large variety of paraphrases suggest that the alternative approach should not be abandoned. The alternative is to index both documents and queries by decomposing complex expressions and constructing simpler descriptors of a normalized form. The syntactic phrase indexing method of chapter 3 takes this approach.

Sparck Jones and Tait applied the variant generation procedure to 10 queries and processed them against a collection of 11,429 abstracts. Their purpose in doing this was to demonstrate the feasibility of the overall procedure, rather than to evaluate the merits of this indexing method. Their results show that retrieval effectiveness with phrase descriptors was lower than with single term descriptors. They emphasize, however, that results due to such a small query sample cannot be taken as indicative of the value of the procedure (Sparck Jones and Tait 1984a:60-63).

Lewis and Croft (1987) have done some preliminary work toward extending Croft's approach to incorporating term dependencies into the retrieval process (Croft 1986a).[11] Instead of constructing groups of dependent terms by identifying syntactically correct natural language phrases in the text of queries, a frame-based representation of query content is used. Rather than representing a query as an unstructured set of word stems, their representation language is based on a controlled vocabulary of concepts designed to be appropriate for scientific and technological fields in general. These concepts are represented as frames that also provide for the specification of relationships among a set of frames used to represent a query.

For the experiments discussed in Lewis and Croft (1987), frame-based representations of 50 queries from the CACM collection were constructed by hand and used as the basis for retrieval experiments. An expectation-based parser is being developed with the objective of constructing query representations automatically. These representations were used both to select individual terms for use in query-document matching, and for identifying groups of

[11] See Croft and Lewis (1987) for an earlier discussion of this work.

dependent terms to modify the document ranking in the same fashion done in Croft (1986a).

From their experimental results, three comparisons are of primary interest here. First, indexing of queries by excluding stopwords was compared to selection of query terms based on the frame representation of query content. For this comparison, it appears that in both cases the document ranking method incorporated the equivalent of tf×idf term weights. Selection of terms based on the frame representation yielded an increase in average precision of 7.6% over indexing simply by removal of stopwords. Second, instead of using tf×idf weights for the frame-based query terms, weights based on the occurrence of the terms in their science lexicon and in a general dictionary were used. This resulted in an increase in average precision of approximately 11% over selection of query terms by stopword removal and use of tf×idf weights. Finally, term dependencies derived from the frame representation were used to modify the document ranking due to single term indexing. This yielded an increase of 3.2% over single terms selected from the frame representations and dictionary-based weights. This is a total increase in average precision of 15.1% over selection of query terms by stopword removal and use of tf×idf weights.

As pointed out by Lewis and Croft, it is important to note that most of the benefit provided by the frame representation appears to be in using it as a basis for selecting query search terms rather than in identifying term dependencies. The use of dictionary information for assigning term weights also appears to be valuable.

Lewis and Croft's results for retrieval using term dependencies are not directly comparable to either the non-syntactic or syntactic phrase indexing experiments of the present study, because their retrieval experiments that involve term dependencies also make use of single terms derived from the frame representations, as well as weights derived from dictionaries. Nevertheless, their 15.1% increase in average precision is somewhat less than the 22.7% increase achieved using the non-syntactic phrase indexing procedure of chapter 2. Their 15.1% increase is somewhat higher than the 8.7% increase achieved using the syntactic phrase indexing procedure of chapter 3. However, only 3-4% of this 15.1% increase can be attributed to the term dependencies derived from the frame representations. Thus it appears that the information about term relationships provided by surface syntactic parsing may be as helpful for purposes of document retrieval as information derived from more complex frame representations.

CHAPTER 5

CONCLUSION

5.1. The Effectiveness of Phrase indexing

Based on average precision figures, the experimental results of chapters 2 and 3 show that under some circumstances phrase descriptors can have a significant positive influence on retrieval performance. If a sufficient number of phrase descriptors are assigned to documents and queries, and if these descriptors are predominantly good indicators of document content and information need, then substantial improvements can be achieved. This is indicated by the 22.7% increase in average precision achieved using non-syntactic phrase indexing on the CACM collection. However, if only a few phrase descriptors are assigned, or if the quality of the descriptors is uneven, then only moderate to slight increases in effectiveness result. This is indicated by the smaller increases in average precision of 11.9%, 8.9%, 4.0%, and 2.2% using non-syntactic phrase indexing on the INSPEC, CRAN, MED, and CISI collections, and the 8.7% and 1.2% increases using syntactic phrase indexing on CACM and CISI.

With regard to the relative value of non-syntactic and syntactic phrase indexing, the precision averages show that non-syntactic phrase indexing is significantly better than syntactic phrase indexing for the CACM collection, but that the difference is insignificant for the CISI collection. Examination of individual queries, however, shows that there is great variability in the performance of both syntactic and non-syntactic phrase indexing. Further, the CISI collection provides some evidence suggesting that syntax-based phrase indexing offers some benefits over non-syntactic phrase indexing. That is, when syntactic phrases are used, the performance of more queries improves over single term indexing than when non-syntactic phrases are used. In addition, 22 queries perform better with syntactic phrases than with non-syntactic phrases, whereas only 7 queries perform better with non-syntactic phrases than with syntactic phrases.

Analysis of the performance of individual queries also revealed some strengths and weaknesses of both phrase indexing methods. The primary strength of non-syntactic phrase indexing appears to be the unrestrictive nature of the method. As shown in section 4.1, this characteristic makes it possible for the non-syntactic method to identify many phrases that could not be recognized when syntactic relationships among words are taken into consideration. When the nature of the text is such that unrestricted word combinations yield good content indicators (for example, with short, well-focused queries), then this characteristic of the method is beneficial. This characteristic, however, also appears to be a serious weakness of the method. When documents and queries are longer, such an unrestrictive approach yields many undesirable phrases. A further short-coming of the method as it is currently implemented is that it can be made more selective only by adjusting the frequency and proximity parameters. When made very restrictive on this basis (as for the CISI collection), the overall effect of phrase indexing is very small because only a few phrase descriptors are assigned to each query and document.

The primary strength of syntactic phrase indexing is its selectivity. By taking into consideration the syntactic structure of text, many undesirable phrase descriptors can be avoided. Examples and their effects on document ranking appear in section 4.1. The ability to identify important relationships among words at fairly long distances, while avoiding the construction of phrases from unrelated words at closer proximities, is also an advantage. This is evidence that it can be beneficial to take text structure into consideration when constructing complex descriptors. A serious shortcoming of the syntactic method is that its selectivity results in the assignment of a relatively small number of descriptors, so the net effect of phrase indexing is small.

There is potential for improvement of both methods. Greater selectivity could be introduced into the non-syntactic method in several ways. One possibility would be to use dictionary information to exclude certain words from use as phrase elements. This could be done on the basis of word classes, or use of an extended stoplist. Further benefit could be derived from being more selective in combining words into phrases, and in regularizing the order of phrase elements. This could be done, for example, by placing limits on the number of stopwords that may intervene between phrase elements, treating conjunctions differently from other non-content words, changing the order of phrase elements only if a stopword (e.g., a preposition) intervenes, and taking into consideration punctuation between potential phrase elements.[1] Improvements of this kind, however, are clearly moving in the direction of simplified syntax, since some information about word classes is involved, and relationships among words are identified on the basis of text characteristics other than simple proximity. Progressive refinement of improvements along these lines would almost certainly become increasingly syntactic in orientation.

There appears to be much greater potential for improvement and extension of the syntax-based approach to phrase indexing. This is the topic of the next section.

5.2. Refinements and Extensions of Syntax-based Indexing

Refinements of syntactic phrase indexing should concentrate on the selectivity of the method. As indicated by the examples discussed in chapter 4, the selectivity offered by syntactic phrase indexing appears to be beneficial in most cases. However, in order for this method to have a greater effect on retrieval performance, more phrases must be generated while at the same time maintaining an appropriate level of selectivity. The existing implementation could be modified in several ways to accomplish this.

---

[1] For further discussion, see section 2.4.1.

Currently only nominal constructions (noun phrases and prepositional phrases) are treated in detail. Among verbal constructions, only infinitival and participial clauses are used as sources of phrase descriptors. By extending the phrase construction rules to include a comprehensive treatment of verbal constructions, many more useful phrase descriptors could be generated (see section 3.5.2 for an example).

The present treatment of adjective phrases has intentionally been made very selective in order to avoid constructing many inappropriate phrases. It is clearly too restrictive, however. For example, by allowing only adjective phrases with participles as heads to be the source of phrase descriptors, some good phrases are lost. One example is *computationally intractable*. The phrase indexing rules could be extended to selectively include constructions of this type. This would necessitate compiling further lexical subclasses, however, in order to avoid using certain low-content words as both heads and modifiers.[2]

The strategy discussed in section 3.4.6.1 of replacing certain semantically general heads of complex noun phrases with a premodifier, has proven to be a useful method of increasing the number of good phrases identified. This approach can be extended, however, to yield additional useful descriptors. Currently, this strategy is applied only to heads of constructions. Further

---

[2] **For example, evaluative words such as** *admirable, admirably, considerable, considerably, enjoyable, notable, notably, preferable,* **and** *preferably* **should be avoided.**

benefit could be realized by also applying it to heads of modifying constructions. An example indicating the usefulness of this extension appears in section 4.1. In that example, the phrase indexing rules fail to construct *retrieval evaluation* from *evaluation of retrieval systems*, since *systems* (not *retrieval*) is a modifier of *evaluation*.

The class of semantically general (or semantically empty) nouns could be employed to improve the analysis of complex noun phrases, which could in turn have a positive effect on phrase indexing. For example, in phrases like *information system architecture*, nominal modifiers preceding a general noun like *system* should most often be analyzed as modifiers of the general noun, rather than as modifiers of the head of the entire noun phrase, in this case, *architecture*. This kind of information could be used at various steps of analysis: (a) directly in the grammar, to aid in disambiguating the syntactic structure of the noun phrase, (b) in calculation of the parse metric, to assign greater value to the parse having this structure, (c) as a post-processing step to adjust the final parse tree, or (d) as part of the phrase construction process.

More selective use of prepositional phrases as postmodifiers of nouns may yield improvements in the quality of phrases. Subject to other constraints on phrase construction, presently all prepositional phrases are potential sources of modifiers of nouns. It may be possible to avoid constructing some undesirable phrases by excluding some prepositional phrases based on the preposition that the phrase contains, or based on the preposition together with the head of the noun phrase object of the prepositional phrase (see section 3.5.2).

In addition to refinement of the phrase construction rules, phrase indexing could also benefit from refinement of the syntactic analysis system. One step that could be taken to improve the quality of syntactic parsing would be to take into consideration information about the complement structure of verbs. This would help to do a better job of prepositional phrase attachment. For example, if a grammar had access to information indicating that the verb *submit* typically has a prepositional phrase complement with *to* as the preposition (*submit* NP_1 *to* NP_2), it would be possible to avoid attaching the prepositional phrase as a modifier of the first noun phrase, rather than as a modifier of the verb.

Much better use could be made of hyphenated forms as sources of phrases if they could be analyzed syntactically and then incorporated into the parses of the sentences in which they occur. Presently, the parser provides no information about the internal syntactic structure of hyphenated forms.

Though the parse metric provided by the PLNLP system is a very useful facility for an application like document content analysis, it may be possible to enhance its usefulness by tailoring the evaluation procedure to behave differently for different kinds of constructions. The parses in (3.70) and (3.71) of section 3.5 provide an example. In this case, the construction *not only ... but also* provides information that could be used to help identify (3.71) as the preferred parse. This information could be taken into consideration in calculating the parse metric in order to give this parse a better rank. Refinement of this parse ranking strategy could itself be the subject of interesting and substantive research, the results of which would be useful not only for applications like document content analysis, but also for the field of syntactic parsing in general.

Aside from direct refinement of the phrase identification process, other approaches to increasing the number of good phrase descriptors should also be examined. One approach would be to use the collection dictionary as a source of information for query expansion. A problem noted in section 4.1 was that a number of good phrases may be identified in a query, but that often only a few of these phrases occur in the documents. It would be useful to expand the query by adding to it phrases that occur in documents and that are closely related to the content of the query, even though they were not identified in the text of the query. This can be done by finding phrases in the collection dictionary that contain words occurring in the query, and then adding such phrases to the query. An example of such a situation would be a query containing the phrase descriptor *algorithm complexity* (from the text phrase *complexity of algorithms*), but not *computational complexity*. A phrase match would not result between this query and a document containing only *computational complexity*. Given the single terms in the query, phrases in the collection dictionary containing *complexity* could be extracted and presented to the user for inspection. The user could then select phrases from this list to be added to the query. This expanded query would then be used as the basis for retrieval. This approach to query expansion is based on a similar method implemented by Hillman (1968).

A simplified version of this strategy has been implemented and tested experimentally with non-syntactic phrases (Scott 1986). The major problem that appeared in these experiments was that far too many phrases were identified. It is not realistic to expect a user to be able to successfully select a few additional useful phrases from an extensive list of candidates. Here again, the problem is one of selectivity. This approach may be more successful if used in conjunction with the syntax-based phrase indexing method rather than the non-syntactic method, since many fewer phrases would be identified. Further refinement could be gained if a thesaurus could be used to help reduce the list of candidate phrases presented to the user. This approach would be especially helpful, for example, with a query that contains a phrase like *syntactic analysis*, and documents that contain only *grammatical analysis*. A list of all phrases containing *analysis* as head would be far too long to present to the user for inspection, because *analysis* has a high frequency of occurrence. A thesaurus could be used, however, to reduce the list to phrases related to language and linguistics, which would be more manageable.

The issue of phrase weighting could also be examined more extensively. Smeaton (1987) has done some experimentation with assigning different weights to groups of dependent terms depending on how the individual terms are related in the source text. For example, he found that effectiveness could be improved by giving more weight to pairs consisting of a head and modifier than to pairs consisting of two heads. This general idea could be extended to take into consideration whether a phrase comes from an unambiguous parse, a fitted parse, or a syntactically ambiguous construction. A phrase weight could be reduced or increased in accordance with the probable degree of accuracy of the parse from which it is taken.

The immediate objective of this study has been to develop and test ways of using information about the syntactic structure of text to construct phrase descriptors that are good indicators of document content. This immediate objective, however, is viewed as part of a more general goal, namely, the development of ways of incorporating information about text structure in general into the overall process of document content analysis. The next stage of development, then, should go beyond the use of syntactic and simple lexical information in analyzing text structure, and should involve the entire content analysis process, not just phrase indexing.

It appears that information about syntactic relationships among words could be used as a basis for being more selective in assigning single terms as descriptors, and also for assigning weights to single term descriptors. For example, query 10 from the CISI collection contains the noun phrase *abstract mathematics*. Several of the non-relevant documents retrieved at a rank of 30 or higher by this query contain the word *abstract* or a derivationally related form like *abstracting*. In all of these non-relevant documents, *abstract* has the sense "short summary" rather than the sense "non-concrete" or "theoretical," as in the query. All of these documents thus have to do with topics like production and evaluation of abstracts, rather than topics related to theoretical endeavors like abstract mathematics.

The negative effects of such matches could be reduced if an indexing procedure could recognize that the word *abstract* should be treated differently when it occurs as a modifier in a phrase like *abstract mathematics* than when it occurs as the head of a phrase like *informative abstracts* or simply *abstracts*.

When occurring as a modifier, a word such as *abstract* could either be rejected as a single term descriptor, or be given a reduced weight, thus eliminating or reducing its effect on query-document similarity values, and potentially lowering the rank of documents that match just on the single term *abstract* and not on the entire phrase *abstract mathematics*. This would also increase the likelihood that documents containing the single term *mathematics* would be retrieved at higher ranks than documents containing the single term *abstract*.

Recent experimental work of Lewis and Croft (1987) indicates that taking information about text structure into consideration in selecting single term descriptors can lead to improvements in retrieval performance (see section 4.2). Further investigation along these lines thus may be of value.

Information about text structure that goes beyond direct syntactic relationships among words can also be incorporated into the content analysis process. One area that could easily be examined within the general approach of the present study would be the development of a set of rules for identifying commonly occurring expressions that are used primarily to provide coherence to text rather than to convey its basic content. Some simple examples of such constructs were discussed briefly in sections 3.4.6.2 and 3.5.2; these include: *a group of*, *in terms of*, and *as a function of*. By recognizing expressions such as these, undesirable single terms and phrases can be avoided.

This approach could be extended to larger constructs, such as common expressions used to introduce queries (for example, *I am interested in information on ...*, *Find documents related to ...*), as well as expressions that introduce the topic of a document, as stated in its abstract (for example, *The objective of this research ...*, *This paper discusses ...*).[3] A systematic attempt to develop rules for identifying expressions that introduce queries and topics could serve as a useful test of this general idea. If this attempt were successful, then the technique could probably be generalized to deal with other kinds of essentially content-less constructions.

The objective of the above approach is to identify expressions that give coherence to discourse but that are low in content, and exclude them from use as content indicators. A similar approach could be taken to use discourse

[3] John Tait (1984) has done some preliminary work in this area.

clues to identify textual elements that are likely to be good content indicators. Much of the work on automatic abstracting and extracting that advocates the use of clue words and expressions for identifying important text passages could be exploited for this purpose (Rush, Salvador, and Zamora 1971; Paice 1981). A further level of refinement would be to use discourse structure to identify relationships among concepts expressed in an abstract that go beyond syntactic relationships, thus adding further precision to the representation of document content (Liddy 1987).

The general approach advocated here is to develop rules that constitute knowledge about the structure of document and query texts, and that specify how that knowledge can be used for purposes of analyzing and representing the content of these texts. The knowledge (or information) these rules contain, however, is general knowledge about syntactic and discourse structure rather than knowledge about a restricted domain. The PLNLP system for natural language processing, together with the SMART system for retrieval experimentation, would provide a convenient environment for implementing and testing additional refinements such as these.

REFERENCES

Aladesulu, O. S. 1985. Improvement of Automatic Indexing through Recognition of Semantically Equivalent Syntactically Different Phrases. Ph.D. Thesis, Ohio State University, Department of Information and Computer Science, Columbus, Ohio.

Apresjan, Ju. D.; Mel'chuk, I. A.; and Zholkovsky, A. K. 1969. Semantics and Lexicography: Towards a New Type of Unilingual Dictionary. In: Ferenc Kiefer, Ed., *Studies in Syntax and Semantics*. D. Reidel, Dordrecht: 1-33.

Baxendale, P. B. 1958. Machine-Made Index for Technical Literature—An Experiment. *IBM Journal of Research and Development* 2:354-361.

Baxendale, P. B. 1961. An Empirical Model for Machine Indexing. In: *Machine Indexing: Progress and Problems*. Third Institute for Information Storage and Retrieval, February 13-17, 1961. Center for Technology and Administration, School of Government and Public Administration, The American University, Washington, D.C.: 207-218.

Boguraev, B. K. 1979. Automatic Resolution of Linguistic Ambiguities. Technical Report No. 11 (Ph.D. Thesis, University of Cambridge), Computer Laboratory, University of Cambridge, Cambridge, England.

Bruandet, M-F. 1987. Outline of a Knowledge Base Model for an Intelligent Information Retrieval System. In: C. T. Yu and C. J. van Rijsbergen, Eds., *Proceedings of the Tenth Annual International ACMSIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, New York: 33-43.

Buckley, C. 1985. Implementation of the SMART Information Retrieval System. Technical Report TR85-686, Department of Computer Science, Ithaca, New York.

Buckley, C. and Lewit, A. F. 1985. Optimization of Inverted Vector Searches. In: *Proceedings of the Eighth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Montreal, Quebec, Canada, June 5-7, 1985. Association for Computing Machinery, New York: 97-110.

Cleverdon, C. W. and Mills, J. 1963. The Testing of Index Language Devices. *Aslib Proceedings* 15(4):106-130.

Cleverdon, C. W.; Mills, J.; and Keen, E. M. 1966. Factors Determining the Performance of Indexing Systems, Vol. 1—Design. Aslib Cranfield Research Project, Cranfield, England.

Climenson, W. D.; Hardwick, N. H.; and Jacobson, S. N. 1961. Automatic Syntax Analysis in Machine Indexing and Abstracting. *American Documentation* 12(3):178-183.

Cooper, W. S. 1984. Bridging the Gap between AI and Information Retrieval. In: C. J. van Rijsbergen, *Research and Development in Information Retrieval: Proceedings of the Third Joint BCS and ACM Symposium,* Kings College, Cambridge, 2-6 July 1984. Cambridge University Press, Cambridge: 259-265.

Cowie, J. R. 1983. Automatic Analysis of Descriptive Texts. In: *Proceedings of the Conference on Applied Natural Language Processing,* 1-3 February 1983, Santa Monica, California. Association for Computational Linguistics: 117-123.

Croft, W. B. 1986a. Boolean Queries and Term Dependencies in Probabilistic Retrieval Models. *Journal of the American Society for Information Science* 37(2):71-77.

Croft, W. B. 1986b. User-Specified Domain Knowledge for Document Retrieval. In: Fausto Rabitti, Ed., *Proceedings of the 1986 ACM Conference on Research and Development in Information Retrieval,* Pisa, Italy, September 8-10, 1986. Association for Computing Machinery: 201-206.

Croft, W. B. and Lewis, D. D. 1987. An Approach to Natural Language Processing for Document Retrieval. In: C. T. Yu and C. J. van Rijsbergen, Eds., *Proceedings of the Tenth Annual International ACMSIGIR Conference on Research and Development in Information Retrieval.* Association for Computing Machinery, New York: 26-32.

DeJong, G. 1983. Artificial Intelligence Implications for Information Retrieval. In: J. J. Kuehn, Ed., *Proceedings of the Sixth Annual International ACM SIGIR Conference.* Association for Computing Machinery, New York: 10-17.

Dillon, M. and Gray, A. S. 1983. FASIT: A Fully Automatic Syntactically Based Indexing System. *Journal of the American Society for Information Science* 34(2):99-108.

Dillon, M. and McDonald, L. K. 1983. Fully Automatic Book Indexing. *Journal of Documentation* 39(3):135-154.

Di Benigno, M. K.; Cross, G. R.; and deBessonet, C. G. 1986. COREL - A Conceptual Retrieval System. In: Fausto Rabitti, Ed., *Proceedings of the 1986 ACM Conference on Research and Development in Information Retrieval,* Pisa, Italy, September 8-10, 1986. Association for Computing Machinery: 144-148.

Doyle, L. B. 1961. Semantic Road Maps for Literature Searchers. *Journal of the Association for Computing Machinery* 8(4):553-578.

Doyle, L. B. 1962. Indexing and Abstracting by Association. *American Documentation* 13(4):378-390.

Earl, L. L. 1970. Experiments in Automatic Indexing and Extracting. *Information Storage and Retrieval* 6:313-334.

Earl, L. L. 1972. The Resolution of Syntactic Ambiguity in Automatic Language Processing. *Information Storage and Retrieval* 8(6):277-308.

Evslin, T. 1965. A General Discussion. Information Storage and Retrieval, Scientific Report to the National Science Foundation 9: II-1-II-15, Department of Computer Science, Cornell University, Ithaca, New York.

Fagan, J. L. 1985. Using PLNLP for Content Analysis in Information Retrieval. Paper presented at the symposium on PLNLP: The Programming Language for Natural Language Processing at the Annual Meeting of the Linguistic Society of America, Seattle, Washington, December 27-30.

Fagan, J. L. 1987. Automatic Phrase Indexing for Document Retrieval: An Examination of Syntactic and Non-syntactic Methods. In: C. T. Yu and C. J. van Rijsbergen, Eds., *Proceedings of the Tenth Annual International ACMSIGIR Conference on Research and Development in Information Retrieval.* Association for Computing Machinery, New York: 91-101.

Fox, E. A. 1983a. Characterization of Two New Experimental Collections in Computer and Information Science Containing Textual and Bibliographic Concepts. Technical Report TR83-561, Department of Computer Science, Cornell University, Ithaca, New York.

Fox, E. A. 1983b. Extending the Boolean and Vector Space Models of Information Retrieval with P-Norm Queries and Multiple Concept Types. Ph.D. Thesis, Cornell University, Department of Computer Science, Ithaca, New York.

Gardin, J-C. 1973. Document Analysis and Linguistic Theory. *Journal of Documentation* 29(2):137-168.

Giuliano, V. E. 1965. The Interpretation of Word Associations. In: M. E. Stevens, V. E. Giuliano, and L. B. Heilprin, Eds., *Statistical Association Methods for Mechanized Documentation,* Symposium Proceedings, Washington, 1964. National Bureau of Standards Miscellaneous Publication 269: 25-32.

Giuliano, V. E. and Jones, P. E. 1963. Linear Associative Information Retrieval. In: P. W. Howerton and D. C. Weeks, Eds., *Vistas in Information Handling,* Volume I: *The Augmentation of Man's Intellect by Machine.* Spartan Books, Washington, D.C.: 30-54.

Hahn, U. and Reimer, U. 1985. The TOPIC Project: Text-Oriented Procedures for Information Management and Condensation of Expository Texts, Final Report. Bericht TOPIC-17/85, Universitaet Konstanz, Konstanz.

Harper, D. J. and van Rijsbergen, C. J. 1978. An Evaluation of Feedback in Document Retrieval Using Cooccurrence Data. *Journal of Documentation* 34(3):189-206.

Harris, Z. S. 1959. Linguistic Transformations for Information Retrieval. In: *Proceedings of the International Conference on Scientific Information* (1958), 2, NAS-NRC Washington, D.C. Reprinted in *Papers in Structural and Transformational Linguistics,* Z. S. Harris, D. Reidel Publishing Co., Dordrecht-Holland, 1970: 458-471. .

Heidorn, G. E. 1972. Natural Language Inputs to a Simulation Programming System. Technical Report NPS-55HD72101A, Naval Postgraduate School, Monterey, California.

Heidorn, G. E. 1975. Augmented Phrase Structure Grammars. In: R. Schank and B. L. Nash-Webber, Eds., *Theoretical Issues in Natural Language Processing: An Interdisciplinary Workshop in Computational Linguistics, Psychology, Linguistics, and Artificial Intelligence,* 10-13 June 1975: 1-5. .

Heidorn, G. E. 1982. Experience with an Easily Computed Metric for Ranking Alternative Parses. In: *Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics,* Toronto, Canada. Association for Computational Linguistics: 82-84.

Heidorn, G. E.; Jensen, K.; Miller, L. A.; Byrd, R. J.; and Chodorow, M. S. 1982. The EPISTLE Text-Critiquing System. *IBM Systems Journal* 21(3):305-326.

Hillman, D. J. 1968. Negotiation of Inquiries in an On-Line Retrieval System. *Information Storage and Retrieval* 4:219-238.

Hillman, D. J. 1973. Customized User Services via Interactions with LEADERMART. *Information Storage and Retrieval* 9:587-596.

Hillman, D. J. and Kasarda, A. J. 1969. The LEADER Retrieval System. *AFIPS Proceedings* 34:447-455.

Jensen, K. 1986. PEG 1986: A Broad-Coverage Computational Syntax of English. Research Report, IBM Thomas J. Watson Research Center, Yorktown Heights, New York.

Jensen, K. 1987. Binary Rules and Non-binary Trees: Breaking Down the Concept of Phrase Structure. In: Alexis Manaster-Ramer, Ed., *Mathematics of Language*. John Benjamins. Forthcoming.

Jensen, K. and Binot, J-L. 1987. Disambiguating Prepositional Phrase Attachments by Using On-line Dictionary Definitions. *Computational Linguistics*. Forthcoming.

Jensen, K. and Heidorn, G. E. 1983. The Fitted Parse: 100% Parsing Capability in a Syntactic Grammar of English. In: *Proceedings of the Conference on Applied Natural Language Processing,* 1-3 February 1983, Santa Monica, California. Association for Computational Linguistics: 93-98.

Jensen, K.; Heidorn, G. E.; Miller, L. A.; and Ravin, Y. 1983. Parse Fitting and Prose Fixing: Getting a Hold on Ill-formedness. *American Journal of Computational Linguistics* 9(3-4):147-160.

Jones, S. and Sinclair, J. McH. 1974. English Lexical Collocations. *Cahiers de Lexicologie* 24(2):15-61.

Klingbiel, P. H. 1973a. Machine-Aided Indexing of Technical Literature. *Information Storage and Retrieval* 9(2):79-84.

Klingbiel, P. H. 1973b. A Technique for Machine-Aided Indexing. *Information Storage and Retrieval* 9(9):477-494.

Klingbiel, P. H. and Rinker, C. C. 1976. Evaluation of Machine-Aided Indexing. *Information Processing and Management* 12(6):351-366.

Knaus, R. 1983. Methods and Problems in Coding Natural Language Survey Data. In: *Proceedings of the Joint Statistical Meetings,* Toronto, Ontario, Canada, August 1983. .

Kuno, S. and Oettinger, A. G. 1962. Multiple-Path Syntactic Analyzer. In: *Proceedings of the IFIP Congress-62*. North Holland.

Langendoen, D. T. and Barnett, H. M. 1986. PLNLP: A Linguist's Introduction. Photocopy, IBM T. J. Watson Research Center, Yorktown Heights, New York.

Lebowitz, M. 1983. Intelligent Information Systems. In: J. J. Kuehn, Ed., *Proceedings of the Sixth Annual International ACM SIGIR Conference.* Association for Computing Machinery, New York: 25-30.

Lesk, M. E. 1969. Word-Word Associations in Document Retrieval Systems. *American Documentation* 20(1):27-38.

Lesk, M. and Evslin, T. 1964. Statistical Phrase Processing. Information Storage and Retrieval, Scientific Report to the National Science Foundation 7: IX-1-IX-10, Department of Computer Science, Cornell University, Ithaca, New York.

Lewis, D. D. and Croft, W. B. 1987. Meaning Representation and Natural Language Processing for Document Retrieval. Photocopy, Computer and Information Science Department, University of Massachusetts, Amherst, Massachusetts.

Liddy, E. D. 1987. Discourse-level Structure in Abstracts. In: *Proceedings of the 50th ASIS Annual Meeting.* Knowledge Industry Publications. Forthcoming.

Liddy, E.; Bonzi, S.; Katzer, J.; and Oddy, E. 1987. A Study of Discourse Anaphora in Scientific Abstracts. *Journal of the American Society for Information Science* 38(4):255-261.

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

Lyons, J. 1968. *Introduction to Theoretical Linguistics.* Cambridge University Press, Cambridge.

Mel'chuk, I. A. 1981. Meaning-Text Models: A Recent Trend in Soviet Linguistics. *Annual Review of Anthropology* 10:27-62.

Mel'chuk, I. A. and Zholkovsky, A. K. 1970. Towards a Functioning 'Meaning-Text' Model of Language. *Linguistics* 56:10-47.

Melton, J. S. 1966. Automatic Language Processing for Information Retrieval: Some Questions. *Proceedings of the American Documentation Institute* 3:255-263.

Metzler, D. P.; Noreault, T.; Richey, L.; and Heidorn, B. 1984. Dependency Parsing for Information Retrieval. In: C. J. van Rijsbergen, Ed., *Research and Development in Information Retrieval: Proceedings of the Third Joint BCS and ACM Symposium,* Kings College, Cambridge, 2-6 July 1984. Cambridge University Press, Cambridge: 313-324.

Miller, L. A. 1980. Project EPISTLE: A System for the Automatic Analysis of Business Correspondence. In: *Proceedings of the First Annual Conference on Artificial Intelligence.* Stanford University: 280-282.

Miller, L. A.; Heidorn, G. E.; and Jensen, K. 1981. Text-critiquing with the EPISTLE system: An Author's Aid to Better Syntax. *AFIPS Conference Proceedings* 50:649-655.

Montgomery, C. A. 1972. Linguistics and Information Science. *Journal of the American Society for Information Science* 23(3):195-219.

Neufeld, M. L.; Graham, L. L.; and Mazella, A. 1974. Machine-Aided Title Word Indexing for a Weekly Current Awareness Publication. *Information Storage and Retrieval* 10(11/12):403-410.

Noreault, T.; McGill, M.; and Koll, M. B. 1981. A Performance Evaluation of Similarity Measures, Document Term Weighting Schemes and Representations in a Boolean Environment. In: R. N. Oddy, S. E. Robertson, C. J. van Rijsbergen, P. W. Williams, Eds., *Information Retrieval Research,* Proceedings of the symposium: Research and Development in Information Retrieval, St. Johns College, Cambridge, June 1980, Joint BCS and ACM Symposium on Information Storage and Retrieval. Butterworths, London: 57-76.

Olney, J.; Lam, V.; and Yearwood, B. 1976. A New Technique for Detecting Patterns of Term Usage in the Text Corpora. *Information Processing and Management* 12(4):235-250.

Paice, C. D. 1981. The Automatic Generation of Literature Abstracts: An Approach Based on the Identification of Self-Indicating Phrases. In: R. N. Oddy, S. E. Robertson, C. J. van Rijsbergen, P. W. Williams, Eds., *Information Retrieval Research,* Proceedings of the symposium: Research and Development in Information Retrieval, St. Johns College, Cambridge, June 1980, Joint BCS and ACM Symposium on Information Storage and Retrieval. Butterworths, London: 172-191.

Paice, C. D. and Aragón-Ramírez, V. 1985. The Calculation of Similarities between Multi-Word Strings Using a Thesaurus. In: *Proceedings of RIAO 1985.* Grenoble, France, 18-20 March 1985: 293-319.

Reeker, L. H.; Zamora, E. M.; and Blower, P. E. 1983. Specialized Information Extraction: Automatic Chemical Reaction Coding from English Descriptions. In: *Proceedings of the Conference on Applied Natural Language Processing,* 1-3 February 1983, Santa Monica, California. Association for Computational Linguistics: 109-116.

Richardson, S. D. 1985. Enhanced Text Critiquing Using a Natural Language Parser. Photocopy, IBM T. J. Watson Research Center, Yorktown Heights, NY.

Rush, J. E.; Salvador, R.; and Zamora, A. 1971. Automatic Abstracting. II. Production of Indicative Abstracts by Application of Contextual Inference and Syntactic Coherence Criteria. *Journal of the American Society for Information Science* 22(4):260-274.

Sager, N. 1975. Sublanguage Grammars in Science and Information Processing. *Journal of the American Society for Information Science* 26(1):10-16.

Sager, N. 1981. *Natural Language Information Processing: A Computer Grammar of English and Its Applications.* Addison-Wesley Publishing Co., Inc., Reading, Massachusetts.

Salton, G. 1962. Manipulation of Trees in Information Retrieval. *Communications of the Association for Computing Machinery* 5(2):103-114.

Salton, G. 1966. Automatic Phrase Matching. In: D. G. Hays, Ed., *Readings in Automatic Language Processing.* American Elsevier, New York: 169-188.

Salton, G. 1968. *Automatic information storage and retrieval.* McGraw-Hill, New York.

Salton, G. 1971. Cluster Search Strategies and the Optimization of Retrieval Effectiveness. In: G. Salton, Ed., *The SMART Retrieval System—Experiments in Automatic Document Processing.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey: 223-242.

Salton, G. 1972a. A New Comparison between Conventional Indexing (MEDLARS) and Automatic Text Processing (SMART). *Journal of the American Society for Information Science* 23(2):75-84.

Salton, G. 1972b. Experiments in Automatic Thesaurus Construction for Information Retrieval. In: C. V. Freiman, Ed., *Information Processing 71: Proceedings of the IFIP Congress 71.* North Holland Publishing Co., Amsterdam: 115-123.

Salton, G. 1975a. *Dynamic Information and Library Processing.* Prentice-Hall Inc., Englewood Cliffs, New Jersey.

Salton, G. 1975b. *A Theory of Indexing.* Regional Conference Series in Applied Mathematics. 18. Society for Industrial and Applied Mathematics, Philadelphia.

Salton, G. 1981. The Smart Environment for Retrieval System Evaluation—Advantages and Problem Areas. In: K. Sparck Jones, Ed., *Information Retrieval Experiment.* Butterworths, London: 316-329.

Salton, G.; Buckley, C.; and Yu, C. T. 1983. An Evaluation of Term Dependence Models in Information Retrieval. In: G. Salton and H-J. Schneider, Eds., *Research and Development in Information Retrieval: Proceedings of the SIGIR/ACM Conference,* Berlin, May 18-20, 1982. Lecture Notes in Computer Science. 146. Springer-Verlag, Berlin: 151-173.

Salton, G. and Lesk, M. E. 1965. The SMART Automatic Document Retrieval System—An Illustration. *Communications of the Association for Computing Machinery* 8(6):391-398.

Salton, G. and Lesk, M. E. 1968. Computer Evaluation of Indexing and Text Processing. *Journal of the Association for Computing Machinery* 15(1):8-36.

Salton, G. and Lesk, M. E. 1971. Information Analysis and Dictionary Construction. In: G. Salton, Ed., *The SMART Retrieval System—Experiments in Automatic Document Processing.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey: 115-142.

Salton, G. and McGill, M. J. 1983. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York.

Salton, G. and Wong, A. 1976. On the Role of Words and Phrases in Automatic Text Analysis. *Computers and the Humanities* 10(2):69-87.

Salton, G.; Wong, A.; and Yang, C. S. 1975. A Vector Space Model for Automatic Indexing. *Communications of the Association for Computing Machinery* 18(11):613-620.

Salton, G. and Yang, C. S. 1973. On the Specification of Term Values in Automatic Indexing. *Journal of Documentation* 29(4):351-372.

Salton, G.; Yang, C. S.; and Yu, C. T. 1974. Contributions to the Theory of Indexing. In: *Information Processing 74*. North Holland Publishing Co., Amsterdam: 584-590.

Salton, G.; Yang, C. S.; and Yu, C. T. 1975. A Theory of Term Importance in Automatic Text Analysis. *Journal of the American Society for Information Science* 26(1):33-44.

Schank, R. C.; Kolodner, J. L.; and DeJong, G. 1981. Conceptual Information Retrieval. In: R. N. Oddy, S. E. Robertson, C. J. van Rijsbergen, P. W. Williams, Eds., *Information Retrieval Research,* Proceedings of the symposium: Research and Development in Information Retrieval, St. Johns College, Cambridge, June 1980, Joint BCS and ACM Symposium on Information Storage and Retrieval. Butterworths, London: 94-116.

Scott, M. A. 1986. The Effect of Term Phrase Refinements on Information Retrieval. Research Project Report, Cornell University, Department of Computer Science, Ithaca, New York.

Shapiro, G. 1965. Statistical Phrase Processing. Information Storage and Retrieval, Scientific Report to the National Science Foundation 9: VII-1-VII-13, Department of Computer Science, Cornell University, Ithaca, New York.

Smeaton, A. F. 1986. Incorporating Syntactic Information into a Document Retrieval Strategy: an Investigation. In: Fausto Rabitti, Ed., *Proceedings of the 1986 ACM Conference on Research and Development in Information Retrieval,* Pisa, Italy, September 8-10, 1986. Association for Computing Machinery: 103-113.

Smeaton, A. F. 1987. Incorporating Syntactic Processing into Document Retrieval. Special Colloquium, August 25, 1987, Cornell University, Department of Computer Science, Ithaca, New York.

Smeaton, A. F. and van Rijsbergen, C. J. 1981. The Nearest Neighbor Problem in Information Retrieval: An Algorithm Using Upper Bounds. *ACM SIGIR Forum* 16(1):83-87.

Sparck Jones, K. 1972. A Statistical Interpretation of Term Specificity and Its Application to Retrieval. *Journal of Documentation* 28:11-20.

Sparck Jones, K. 1974. Automatic Indexing. *Journal of Documentation* 30(4):393-432.

Sparck Jones, K., Ed. 1981. *Information Retrieval Experiment.* Butterworths, London.

Sparck Jones, K. and Kay, M. 1973. *Linguistics and Information Science.* Academic Press, New York.

Sparck Jones, K. and Kay, M. 1977. Linguistics and Information Science: A Postscript. In: D. E. Walker, A. H. Karlgren, and M. Kay, Eds., *Natural Language in Information Science: Perspectives and Directions for Research.* Skriptor, Stockholm: 183-192.

Sparck Jones, K. and Tait, J. I. 1984a. Automatic Search Term Variant Generation. *Journal of Documentation* 40(1):50-66.

Sparck Jones, K. and Tait, J. I. 1984b. Linguistically Motivated Descriptive Term Selection. In: *Proceedings of COLING 84,* 2-6 July 1984, Stanford University, California. Association for Computational Linguistics: 287-290.

Steinacker, I. 1973. Some Aspects of Computer Text Processing. *Data Processing* 15(2, 3):86-88, 148-153.

Steinacker, I. 1974. Indexing and Automatic Significance Analysis. *Journal of the American Society for Information Science* 25(4):237-241.

Stiles, H. E. 1961. The Association Factor in Information Retrieval. *Journal of the Association for Computing Machinery* 8(2):271-279.

Strong, S. M. 1973. An Algorithm for Generating Structural Surrogates of English Text. Technical Report OSU-CISRC-TR-73-3 (M.S. Thesis), Computer and Information Science Research Center, The Ohio State University, Columbus, Ohio.

Strong, S. M. 1974. An Algorithm for Generating Structural Surrogates of English Text. *Journal of the American Society for Information Science* 25(1):10-24.

Tait, J. I. 1984. Automatic request parsing and variant generation. In: K. P. Jones, Ed., *Intelligent Information Retrieval: Informatics 7.* Aslib, London: 53-63.

Tuttle, M. S.; Sherertz, D. D.; Blois, M. S.; and Nelson, S. 1983. Expertness from Structured Text?: RECONSIDER: A Diagnostic Prompting Program. In: *Proceedings of the Conference on Applied Natural Language Processing,* 1-3 February 1983, Santa Monica, California. Association for Computational Linguistics: 124-131.

van Rijsbergen, C. J. 1977. A Theoretical Basis for the Use of Cooccurrence Data in Retrieval. *Journal of Documentation* 33:106-119.

Vickery, A.; Brooks, H.; and Robinson, B. 1987. A Reference and Referral System Using Expert System Techniques. *Journal of Documentation* 43(1):1-23.

Vladutz, G. 1983. Natural Language Segmentation Techniques Applied to the Automatic Compilation of Printed Subject Indexes and for Online Database Access. In: *Proceedings of the Conference on Applied Natural Language Processing,* 1-3 February 1983, Santa Monica, CA. Association for Computational Linguistics: 136-142.

Vladutz, G. and Garfield, E. 1979. KWPSI—An Algorithmically Derived Key Word/Phrase Subject Index. In: R. D. Tally and R. R. Deultgen, Eds., *Information Choices and Policies.* Proceedings of the ASIS Annual Meeting, Minneapolis, Minnesota, October 14-18, 1979. 16. Knowledge Industry Publications, White Plains, New York: 236-245.

Voorhees, E. M. 1985. The Effectiveness and Efficiency of Agglomerative Hierarchic Clustering in Document Retrieval. Technical Report TR85-705 (Ph.D. Thesis), Department of Computer Science, Cornell University, Ithaca, New York.

Waldstein, R. 1981. The Role of Noun Phrases as Content Indicators. Ph.D. Thesis, School of Information Studies, Syracuse University, Syracuse, New York.

Walker, D. E. 1981. The Organization and Use of Information: Contributions of Information Science, Computational Linguistics, and Artificial Intelligence. *Journal of the American Society for Information Science* 32:347-363.

Walker, D. E. and Hobbs, J. R. 1981. Natural Language Access to Medical Text. Technical Note 240, Project 1944, SRI International, Menlo Park, California.

Winograd, T. 1983. *Language as a Cognitive Process:* Vol. 1, *Syntax.* Addison-Wesley, Reading, Massachusetts.

Young, C. E. 1973. Development of Language Analysis Procedures with Application to Automatic Indexing. Technical Report OSU-CISRC-TR-73-2 (Ph.D. Thesis), Computer and Information Science Research Center, The Ohio State University, Columbus, Ohio.

Yu, C. T.; Buckley, C.; Lam, K.; and Salton, G. 1983. A Generalized Term Dependence Model in Information Retrieval. *Information Technology: Research and Development* 2:129-154.

Yu, C. T.; Salton, G.; and Siu, M. K. 1978. Effective Automatic Indexing Using Term Addition and Deletion. *Journal of the Association for Computing Machinery* 25(2):210-225.

Zholkovsky, A. K. and Mel'chuk, I. A. 1970. Semantic Synthesis. *Systems Theory Research* 19:170-243.