An Approach to Natural Language Processing for Document Retrieval

W. Bruce Croft
David D. Lewis
Computer and Information Science Department
University of Massachusetts, Amherst, MA. 01003

Abstract

Document retrieval systems have been restricted, by the nature of the task, to techniques that can be used with large numbers of documents and broad domains. The most effective techniques that have been developed are based on the statistics of word occurrences in text. In this paper, we describe an approach to using natural language processing (NLP) techniques for what is essentially a natural language problem - the comparison of a request text with the text of document titles and abstracts. The proposed NLP techniques are used to develop a request model based on "conceptual case frames" and to compare this model with the texts of candidate documents. The request model is also used to provide information to statistical search techniques that identify the candidate documents. As part of a preliminary evaluation of this approach, case frame representations of a set of requests from the CACM collection were constructed. Statistical searches carried out using dependency and relative importance information derived from the request models indicate that performance benefits can be obtained.

1 Introduction

Document retrieval is a task that involves the comparison of text or representations of text with representations of users' information needs. The aim of this comparison is to identify documents or parts of documents that address the information need. Current approaches to document retrieval emphasize the use of fairly simple techniques that are based on statistical models of word importance. For example, the probabilistic model [RIJS79] estimates the probability of relevance of a document using Bayesian classification theory. The representations of documents and information needs (requests) that are used for this model are simply sets of unweighted words or index terms. This basic approach can be extended to incorporate weighted terms [CROF81] or requests structured using Boolean operators [CROF86a].

Statistical indexing and retrieval techniques are efficient and are more effective in terms of finding relevant documents than searches based on Boolean queries and exact matching [SALT83b]. The major disadvantage of these techniques is that the absolute level of performance in terms of effectiveness is still quite low. A number of suggestions have been made to address this problem. The design of the I^{3}R system [CROF86c] was based on the observation that many of the retrieval errors made by a document retrieval system were the result of inadequate representations of information needs. The emphasis in the operation of the I^{3}R system is on the identification of important *concepts* or topics in a user's request and the acquisition of knowledge about the domain of the request [CROF86b]. The *request model* that is constructed is used to provide information about important index terms and term dependencies to statistical retrieval strategies. The flexible control mechanism and architecture of the I^{3}R system supports the construction of the request model by making a variety of facilities available to the user.

The I^{3}R approach is one way of tackling what can be regarded as the central issue in information retrieval - the design and acquisition of appropriate representations for documents and information needs. Since both the requests and the contents of documents are usually expressed in natural language, it may be supposed that techniques designed for natural language processing (NLP) would provide better representations than those used by statistical techniques. The difficulty, however, is in finding NLP techniques that can feasibly be used with the large numbers of documents, broad domains, and very limited domain knowledge that characterize the document retrieval task. Despite these constraints, there are reasons to believe that current NLP techniques can be used to improve the effectiveness of document retrieval systems. The two main reasons are as follows;

- Document retrieval is a different task than, for example, story understanding and does not require a complete, unambiguous interpretation of the text passages involved. There is some evidence that even simple NLP techniques can provide useful information for document retrieval systems [SMEA86,THUR86].
- The importance of an individual request and the user's interpretation of the meaning of that request provides an inherent limitation on the NLP involved [SPAR84]. Rather than attempting to process all document texts independent of individual requests, the NLP component of a document retrieval system should be used to analyze a request plus the texts of only those documents that potentially address the particular information needs expressed in that request.

Additionally, while a traditional NLP system must have all the needed domain and linguistic knowledge specified in advance, a document retrieval system can acquire some of this knowledge as needed from a human expert (the user) during a search session.

The approach described in this paper for augmenting document retrieval systems using NLP is based on these points and on the I^3R framework. The initial goal of the ADRENAL system (Augmented Document REtrieval using NAtural Language processing) will be to produce a detailed representation of the information need by using both NLP and interaction with the user to analyze the text of the request. The request model that is constructed will have an integrated representation of concepts expressed in the request, concepts from the domain knowledge acquired from the user, and relationships between those concepts. Parts of this request model will be used to generate information for statistical retrieval strategies that identify potentially interesting documents. The texts of these candidate documents can then be compared to the full request model using NLP techniques. The details of the request model representation and the NLP techniques being proposed are given in the next section. The third section gives an overview of ADRENAL. In the fourth section we describe an experiment designed to test some parts of the proposed system. Representations of a sample query set are constructed using a case frame approach. The information in these representations is used to provide information to search strategies based on the probabilistic model and the results of these searches are compared to those obtained with standard indexed queries.

2 The Approach

2.1 A Representation for Science and Technology

The idea of representing text in terms of concepts or units of meaning other than single words has been a theme of much research in document retrieval (for example, [SPAR74,DILL83]). Usually these units are fragments of the original text, such as noun phrases, which are extracted without the use of sophisticated NLP techniques. The problem with this approach is that all the system knows about these fragments is that their words are somehow related. Since the structure and meaning of the fragments is not understood, they cannot be matched against pieces of text which, while superficially different, have similar meanings. The goal of applying NLP techniques in a document retrieval system, then, is to transform the linguistic structures in the user request and in selected documents into representations of their meaning, thus allowing similar concepts to be recognized in a variety of textual forms.

If an IR system is to be able to compare representations produced by an NLP component, it is necessary for those representations to be constructed using a language with known semantics. For this purpose we are developing a language called REST (Representation for Science and Technology). REST is a frame-based language which allows the content of a document to be represented in terms of a predefined set of basic scientific concepts. As an example of a REST representation consider the following user query:

> "Probabilistic analyses of Quicksort, a divide and conquer algorithm"

The representation of this query as REST frames is shown in figure 1. Each frame has a *type* corresponding to a concept such as STUDY, METHOD, RELATIONSHIP, etc. The specific meaning of a particular instance of a type arises from the slot relations between it and other frames (via the slots marked DEF) and/or by an APPEARANCE (APP) slot which shows what words in the text correspond to this concept. (Note that while a slot is shown as appearing in one frame, it actually represents a link between two frames. So while we only show an IS-A slot in STUDY-2, conceptually there is an IS-A-INVERSE slot in STUDY-1.) The key point to notice is that the raw text of the query has been given an interpretation in terms of classes of actions, entities,

```
(STUDY-1
   APPEARANCE: analysis)
(STUDY-2
   IS-A(DEF): STUDY-1
   ARGUMENT-OF(DEF): RELATIONSHIP-1)
(STUDY-3
   IS-A(DEF): STUDY-2
   INTEREST(DEF): METHOD-3)
(RELATIONSHIP-1
   APP: probabilistic)
(METHOD-1
   APP: algorithm)
(METHOD-2
   IS-A(DEF): METHOD-1
   USES(DEF): ACTION-1)
(METHOD-3
   IS-A(DEF): METHOD-2
   APP: Quicksort)
(ACTION-1
   APP: divide and conquer)
```

Figure 1: A REST representation

and relationships that are understood by the system. The representation both preserves the original words of the query, for use with term-based retrieval methods, and allows more sophisticated comparisons with documents represented in REST form. For instance, consider a document whose title is

> "Empirical results on the behavior of common recursive algorithms"

If we represented the query and document as simple term vectors they would match poorly, having only one term in common. Methods that make use of noun phrases or other linguistic structures would be similarly ineffective. However, if REST representations of the query and document were available, then their similarity would be detectable: both representations would contain a METHOD frame (specialized by "algorithm"), and the system would know about the connection between STUDY (as in the query) and DATA (the "results" in the document title).

Representing queries and documents in REST form does not, of course, guarantee that an effective comparison can be made between them. There are two major factors that limit the extent to which useful comparisons can be made. The first is that the REST language at present has available only a limited number of fairly general scientific concepts. For instance, in the above examples, only the category RELATIONSHIP was available to rep-

resent the terms "probabilistic" and "empirical". Since a wide variety of textual items would be represented as RELATIONSHIPs, the system would only be able to draw rather weak conclusions about the similarity of these two concepts. One solution is to expand the set of primitive concepts (as was done, for instance, in producing STUDY from ACTION) to include a number of different primitives for relationships and properties, which are currently all subsumed under RELATIONSHIP. REST is designed to allow new concepts to inherit properties from pre-existing ones, thus simplifying the expansion of the representation language.

The other factor that limits the ability of a system to compare REST representations is the extent to which it knows anything about the words in the APPEARANCE slots. Drawing on the above examples again, note that "divide and conquer" is represented simply as an ACTION, while "recursive" would be a RELATIONSHIP. As such the system would be unable to draw any conclusions about how well they matched, unless it had some knowledge of the connection between "divide and conquer" and "recursive". Since we do not wish to assume that a general purpose document retrieval system will have this level of domain knowledge, the alternative is to ask the user to provide information about vocabulary items that are likely to be encountered in relevant documents. Since it would be inappropriate to expect the user to build REST structures to represent this information, we instead follow [CROF86b] and allow the user to enter a set of vocabulary terms and indicate whether certain linguistic or thesaurus-like relationships (such as SYNONYM and GENERALIZATION) hold between them. While these relationships will not support the more powerful inferencing and matching procedures that the REST primitives do, they will allow terms to be matched based on something other than surface form.

2.2 NLP techniques

The advantages of the REST representation will be irrelevant unless mechanical techniques can be used to translate queries and, to some degree, documents into it. As mentioned in the introduction, the document retrieval task is a very challenging one for NLP. On the one hand, if we make the reasonable assumption that a general purpose document retrieval system will not have dictionary entries for words from specific technical domains, then we cannot expect to have syntactic information on all, or even most, of the words encountered in document and query texts. This argues against using a parser which analyzes a sentence in terms of an implicitly or explicitly built representation of its syntactic structure. On the other hand, limitations on domain knowledge also limit the effectiveness of more semantics-oriented parsing techniques. Our proposed solution is, as one might expect, a compromise between the two classes of parsers.

The overall model will be that of an expectation-based parser [SCHA75,BIRN81,CULL86] one of the more successful types of semantics-based parsers. (Other names used for this type of parser include "conceptual analyzer", "request-based parser", and "situation-action parser".) Briefly, these parsers associate a *case frame* [BRUC75] with certain words (particularly verbs) in their lexicons. Each case frame represents some real-world action, and contains slots for the various entities that take part in that action. Associated with each case frame is a set of production rules (called *expectations* or *requests*) that suggest where to look in the sentence for other words, or concepts triggered by other words, to fill the slots in the case frame. Thus as the sentence is processed, a frame-based representation of its meaning is both being created, and being used to guide the interpretation of the rest of the sentence. In point of fact, the representation used by our parser will also be the REST representation of the content of the sentence – the design of the primitives in REST has been guided by the intention to use them both for knowledge representation and parsing. For example, the three STUDY frames in Figure 1 would have originated as a single case frame built by the parser on encountering the word "analysis." Expectations associated with that word would search for modifiers like "probabilistic" and for the entity being analyzed ("Quicksort"), as well as looking for other components that are not present in this particular piece of text.

While being guided by the expectation-based model laid out above, we plan to depart from it in several ways. Traditional expectation-based parsers rely heavily on *slot restrictions* – rules about what semantic classes of words or concepts can fill particular slots in the case frames. The difficulty with this in the document retrieval task is that the interesting, domain-specific words won't appear in our lexicon, so the parser will not know anything about their semantic classes. The system can try to get such information from the user on the most important words related to the query, but this is a small fraction of the domain words that the parser will encounter. Our solution to this problem is based on an extension of the concept of a *phrasal lexicon* [BECK75]. The idea is that certain text structures larger than words are used essentially as a unit and occur frequently enough that they should have lexicon entries similar to individual words. Our intention is to represent these phrasal units as small, syntactic, transition net parsers [WOOD70], and to associate the case frames to be used for expectation-based parsing with these phrasal units rather than with individual words, as is the usual method. The phrasal parsers will take advantage of the presence of common phrases and syntactic cues to analyze pieces of a sentence and fill some of the slots of their associated case frames. We believe that such phrasal constructs are used frequently enough in technical prose to make this an appropriate strategy. Once a number of case frames and other REST structures have been instantiated by phrasal parsing, expectation-based parsing will link them together to form a representation of the entire sentence.

Another departure from usual NLP practice is inspired by the fact that the NLP component does not actually need to *understand* a document, but rather only process it enough to decide whether it is relevant to the user query. Therefore the design of our parser emphasizes finding those sections of a text that contain information useful in making a relevance decision, and then parsing only these sections. Furthermore, since a representation of the sentence will be built up incrementally during parsing, we can integrate parsing the sentence with comparing it to the query, and thus increase efficiency by stopping the parse as soon as enough information is available to make a retrieval decision. The fact that the meaning representation is built incrementally also adds to robustness, since if the parse fails due to some difficult text structure, the partially formed representation is still available for matching. This style of parsing is similar to *text skimming* [DEJO79,TAIT82], though research on that subject has not focused on searching for particular concepts.

The above gives the flavor of our approach to applying NLP to the document retrieval task. We discuss this approach further in the next section, which outlines a system design integrating the use of NLP with traditional IR techniques.

3 System overview

In the following we describe the processing that ADRENAL will use in handling a user request. Detailed design and implementation of the system has begun, with some modules being taken from the I^{3}R system.

It might seem that the ideal time to apply NLP techniques to the system's document collection would be before any user queries are processed, to avoid the time pressure of an interactive situation. However, this would force the NLP component to try to build a complete representation of each document's meaning, without any specific information need, and without aid from a user. Such an approach is untenable with current NLP techniques. However, some preprocessing, can be done both to improve indexing, and to save work for later parsing. The indexing task will use some superficial syntactic analysis, aided by one of the large, machine-readable dictionaries that have recently become available [ALSH85].

The next step, processing of the user query, involves using NLP in a fairly traditional mode; i.e. attempting to produce a complete representation of the meaning of the text. As described above, the parsing will be expectation-based, with the expectations being associated with REST frames instantiated by syntactically defined phrasal units. Assuming that the query is being made in an interactive environment, the system try to confirm its interpretation with the user. Besides building a representation of the query, ADRENAL will ask the user for additional vocabulary terms, and for linguistic relationships between these terms and those in the query, as discussed in Section 2.1.

Once the REST representation of the query is produced, NLP techniques could be used to compare the query to every document in the collection, but this would be hopelessly inefficient. Instead, an initial document ranking will be produced by converting the REST frames into sets of index terms to compare, via a statistical retrieval strategy, with the index terms of documents. Information from the REST representation will be used to enhance the effectiveness of the statistical retrieval; preliminary results on this method are reported in Section 4.

Most documents will score so low on the statistical retrieval that they will not be considered further. A few high-scoring ones might be presented to the user without further analysis. For the remainder of the documents (those with moderately high scores on the statistical retrieval) additional natural language processing with the expectation-based parser may be necessary. The sections of a document to be parsed are chosen based on their potential for producing REST frames that could be usefully matched with the representation of the query. For instance, if ADRENAL were seeking documents in response to the example query on Quicksort (see Section 2.1) a sentence containing the words "statistical" and "divide" would be an excellent choice for parsing, to distinguish good matches like "...the statistical properties of techniques that divide a problem into smaller..." from bad matches, such as "...we divide up AI learning methods into three classes: statistical,...".

As the retrieval process proceeds the user will be encouraged to provide additional or corrected vocabulary information based on their judgment of the relevance of the documents being retrieved. The user will decide when the retrieval process should stop, though ADRENAL will attempt to keep him or her informed of its estimation of the potential of the documents remaining to be checked.

4 The Experiment

Given a REST representation of a request, it is relatively straightforward to generate information for a statistical retrieval strategy. The strategy developed from the probabilistic model by Croft [CROF81,CROF86a] can make use of information about the relative importance of terms and about dependencies between terms. This information is derived from the relative importance of slots in case frames and the term groupings that represent concepts. In the absence of information about the importance of terms in the query, the retrieval strategy produces a ranked list of documents according to the following score;

$$\sum T(x_i) \cdot W(x_i) \cdot x_i + A \qquad (1)$$

where x_i is the ith term in the document representative and is either 1 or 0 depending on whether the term is assigned or not, $W(x_i)$ is a weight related to the collection frequency of term i, and $T(x_i)$ is the term significance weight of term i, and A is a correction factor that depends on the presence of dependent groups of words in a particular document. The summation is done over all the terms in the request. In this section, we shall describe techniques for determining, from the REST representation of the request, the dependencies used in A and their relative importance. Note that the dependencies in A can have relative importance weights (in the theory, these are correlation coefficients) but they have not been used in previous work. The request model can also be used to estimate the relative importance of individual terms in the request.

The simplest method for using information from the request model involves taking the terms for the probabilistic request from the REST representation of the query rather than by removing stopwords from the natural language text and using the remaining terms. When this was done to the 32 queries from the Communications of the ACM (CACM) collection which were used in [CROF86a] the average number of terms per request was 9.6, as opposed to 12.6 terms when a standard stoplist is used. The difference comes from the fact that a term is present in the REST representation only if it is important to the meaning of the query. For instance, the word "information" would be present in the REST representation of "articles on information theory" but would not be present in the representation of "give me information on parallel processing." Our hypothesis was that generating the request in this fashion would considerably increase precision.

The relative importance of query terms is based on the presence or absence of the terms in our science vocabulary and/or in a large general purpose dictionary. For the experiments reported here, we chose to assign query term weights as follows:

```
1.0 if the term is not in our science lexicon
    but not in the general-purpose dictionary.
0.8 if the term is in dictionary
    but not in our science lexicon.
0.6 if the term is in our science lexicon
    but not in the dictionary.
0.4 if in the dictionary
    and in our science lexicon.
```

The hypothesis is that the more general the term, the lower the relative importance in the query. This weight is used to modify the weights in equation 1.

The most interesting way to enhance probabilistic retrieval is to use the fact that the REST representation makes apparent term relationships that are not explicit in the text of the query.

We produced by hand REST representations of a set of queries from the CACM collection, and then automatically generated for each query subsets of terms that the REST representation indicated were related conceptually, and which thus should be considered mutually dependent in a probabilistic model. These dependent term groups were then used to modify the rankings of documents retrieved by a probabilistic retrieval, as was done in [CROF86a].

The algorithm used to generate dependent term groups from a REST representation is described below. It is based on generating *clauses* (sets of related frame names) and then replacing the frame names in the clauses with corresponding sets of terms. We say that frame A is a *defining frame* of frame B if A's name appears in a slot in B marked DEF. A is a neighboring frame of B if A's name appears in *any* slot in B.

1. Generate the *text groups* for each frame. One such group corresponds to the filler of each APPEARANCE slot in the frame, with words from a stoplist removed. If the frame has defining frames, then all combinations formed by taking one text group from each defining frame are also text groups for this frame. (The recursion will always terminate unless there a circular definition is present.)

2. Generate the *0-order clauses* for each REST frame. One 0-order clause is always the name of the frame itself. The other, if present, is the set of all names of defining frames for this frame. The 0-order clauses represent those sets of frames which we know to be closely related, because all the frames named in a 0-order clause are part of the definition of a single concept.

3. Generate the *k-order clauses* for each frame, for k = 1 to a desired maximum value. The set of k-order clauses for a frame is the union of the following sets of clauses:

   a. All (k-1)-order clauses from neighboring frames.

   b. All clauses formed by deleting an element from one of the (k-1)-order clauses for the frame.

   c. All clauses formed by replacing an element from one of the (k-1)-order clauses for the frame with a (k-1)-order clause from a frame neighboring the one whose name was replaced.

4. Generate the *term groups* corresponding to each clause by forming all combinations of replacing each frame name in a clause with one of the text groups representing that frame. Discard all term groups which contain no terms or only one term, as well as all duplicate groups.

The intent behind the algorithm is to produce clauses corresponding to strongly dependent groups of terms before producing clauses for less strongly dependent groups. The factors to consider in choosing an algorithm to generate dependent clauses and in choosing a maximum value for k are discussed under Future Work; for these experiments we generated dependent term groups only from 0-, 1-, and 2-order clauses, and eliminated any group containing more than five terms. Also, as in [CROF86a], only the top 100 documents from the initial retrieval were reranked using information from the dependent term groups.

While the above algorithm does well at producing groups of terms that are indeed dependent, not all such groups should have equal impact on the score of a document that matches the group. In particular, we do not want to give as much weight to groups consisting predominantly of general scientific terms. These terms, which make up a *science lexicon* closely associated with the REST "types", have dependencies which are independent of any particular query, and so should not be counted as heavily. To reflect this we set the correlation coefficient for a term group equal to the proportion of terms in the group that are not from the science lexicon. This means groups which contain mostly domain terms will have the most influence on a document's score. Figure 2 shows an example of text groups, clauses, and dependent term groups (with correlation coefficients) for one of the frames from Figure 1.

To test the effectiveness of these various methods we used them in combination with a probabilistic retrieval incorporating inverse document frequency and within document frequency weights. The use of these two weights is equivalent to the tf.idf model [SALT83b,CROF84] which is regarded as one of the best statistical search strategies. If any of these methods provided additional performance gains above and beyond those provided by the best standard methods, this would be an encouraging preliminary result.

In fact, as shown in Tables 1 and 2 all three methods yielded significant improvements. Comparing columns 1 and 2 in either Table 1 or Table 2 shows significant improvements in precision

```
METHOD-2
   text group: {algorithm, divide, conquer}
   0-order clauses: {METHOD-2},
                    {ACTION-1, METHOD-1}
   1-order clauses: {ACTION-1},
                    {METHOD-1},
                    {ACTION-1, METHOD-2},
                    {METHOD-1, METHOD-2}
   dependent term groups:
            {algorithm, divide, conquer}  .66
            {divide, conquer}             1.0
```

Figure 2: Clauses and Dependent Term Groups (with correlation coefficients)

resulting from using the term set based on the REST representation rather than one from stoplist-pruned natural language text. Comparing columns 2 and 3 in either table shows the improvements at almost all recall levels gained by using term significance weights based on lexicon information. Finally, comparing any column in Table 1 with its counterpart in Table 2 shows the improvements gained in all cases by reranking the top 100 documents based on the clauses generated from the REST representation. It should be emphasized that this result comes from a small sample of manually constructed REST representations. The automatic construction of these representations and evaluating them with the full query set of the ACM collection is the aim of our current system development.

5 Future Work

Development of ADRENAL is just beginning, and there are many issues yet to be resolved in how to apply natural language processing technology to document retrieval, especially with respect to analyzing documents. Therefore, the above results are encouraging in that they suggest that significant performance gains will result even if NLP is applied only to the user query.

The use of REST representations for augmenting probabilistic retrieval can certainly be improved. An obvious extension is that when generating a probabilistic request and dependent term groups from a REST representation, not only those general science terms that appear in the text of the query should be used, but also other science lexicon terms that correspond to the same REST primitives. In effect we would be using the REST concepts as indices into a thesaurus. Term significance weighting could be used to avoid negative effects of adding multiple general terms.

We are also investigating improvements in the algorithm for generating dependent term groups. Heuristics which take advantage of the semantics of the links between frames have the potential of both improving the quality of clauses generated and for making clause generation more efficient.

Acknowledgments

This research was supported in part by NSF grants IST-8414486 and by a NSF Graduate Fellowship. We would like to thank Wendy Lehnert and Daniel Suthers for their advice on the parser, and on REST, respectively.

	Precision		
Recall	Standard term set	REST-derived term set	REST-derived terms + weights
10	45.8	52.3	50.7
20	34.6	35.8	40.9
30	29.1	31.2	34.0
40	27.2	28.1	29.8
50	19.6	21.7	26.0
60	14.5	17.0	20.6
70	9.3	10.5	13.0
80	6.6	7.0	9.3
90	2.7	2.8	3.4
100	1.0	1.0	1.2

Table 1: Probabilistic retrieval using within document frequency weights (preliminary results from 6 queries)

	Precision		
Recall	Standard term set	REST-derived term set	REST-derived terms + weights
10	51.9	52.7	53.2
20	41.1	42.5	47.4
30	30.8	32.1	35.6
40	27.8	28.4	31.8
50	22.7	24.8	28.1
60	16.4	19.0	23.5
70	9.4	10.5	14.3
80	6.6	7.0	9.3
90	2.7	2.8	3.4
100	1.0	1.0	1.2

Table 2: After reranking using dependent term groups (preliminary results from 6 queries)

References

[ALSH85] Alshawi, H.; Boguraev, B.; Briscoe, T. "Towards a Dictionary Support Environment for Real Time Parsing". Technical Report, Computer Laboratory, University of Cambridge, 1985.

[BECK75] Becker, J. D. "The Phrasal Lexicon". Bolt, Beranek, and Newman Inc. Report No. 3081, May 1975.

[BIRN81] Birnbaum, L.; Selfridge, M. "Conceptual Analysis of Natural Language." In *Inside Computer Understanding : Five Programs Plus Miniatures.* Edited by R. Schank and C. Riesbeck, 318-353. Hillsdale : Lawrence Erlbaum, 1981.

[BRUC75] Bruce, B. "Case Systems for Natural Language." *Artificial Intelligence,* 6: 327-360; 1975.

[CROF81] Croft, W. B. "Document Representation in Probabilistic Models of Information Retrieval". *Journal of the American Society of Information Science,* 32: 451-457; 1981.

[CROF84] Croft, W.B. "A Comparison of the Cosine Correlation and the Modified Probabilistic Model". *Information Technology,* 2: 113-114; 1984.

[CROF86a] Croft, W. B. "Boolean Queries and Term Dependencies in Probabilistic Retrieval Models". *Journal of the American society for Information Science,* 37: 71-77; 1986.

[CROF86b] Croft, W.B. "User-Specified Domain Knowledge for Document Retrieval". *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval,* 201-206, Pisa, Italy, 1986.

[CROF86c] Croft, W. B.; Thompson, R. "I^3R: A New Approach to the Design of Document Retrieval Systems". journal of the American Society for Information Science, (to appear).

[CULL86] Cullingford, Richard E. *Natural Language Processing: A Knowledge-Engineering Approach.* Totowa : Rowman & Littlefield, 1986.

[DEJO79] De Jong, G.F. "Skimming Stories in Real Time: An Experiment in Integrated Understanding." Research Report 158, Yale University Department of Computer Science, New Haven, Connecticut, 1979.

[DILL83] Dillon, M.; Gray, A.S. "FASIT: A fully automatic syntactically based indexing system." *Journal of the American Society for Information Science.* 34:99-108; 1983.

[RIJS79] Van Rijsbergen, C. J. *Information Retrieval.* Second Edition. Butterworths, London; 1979.

[SALT83b] Salton, G.; Fox, E.A.; Wu, H. "Extended Boolean information retrieval." *Communications of the ACM.* 26:1022-1036; 1983.

[SCHA75] Schank, R. C., ed. *Conceptual Information Processing.* Amsterdam : North Holland, 1975.

[SMEA86] Smeaton, A.F. "Incorporating Syntactic Information into a Document Retrieval Strategy: An Investigation." *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval*, 103-113, Pisa, Italy, 1986.

[SPAR74] Sparck Jones, K. "Automatic Indexing". *Journal of Documentation, 30*: 393-432; 1974.

[SPAR84] Sparck Jones, K.; Tait, J. I. "Automatic Search Term Variant Generation". *Journal of Documentation,* 40: 50-66; 1984.

[TAIT82] Tait, J.I. "Automatic Summarizing of English Texts." Technical Report 47, University of Cambridge Computer Laboratory, Cambridge, England, 1982.

[THUR86] Thurmair, G. "REALIST: Retrieval Aids by Linguistics and Statistics." *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval*, 138-143, Pisa, Italy, 1986.

[WOOD70] Woods, W. A. "Transition Network Grammars for Natural Language Analysis." *Communications of the ACM.* 13:591-606; 1970.