The Use of Phrases and Structured Queries in Information Retrieval

W. Bruce Croft, Howard R. Turtle[*] and David D. Lewis[†]
Computer and Information Science Department
University of Massachusetts, Amherst, MA 01003

Abstract

Both phrases and Boolean queries have a long history in information retrieval, particularly in commercial systems. In previous work, Boolean queries have been used as a source of phrases for a statistical retrieval model. This work, like the majority of research on phrases, resulted in little improvement in retrieval effectiveness. In this paper, we describe an approach where phrases identified in natural language queries are used to build structured queries for a probabilistic retrieval model. Our results show that using phrases in this way can improve performance, and that phrases that are automatically extracted from a natural language query perform nearly as well as manually selected phrases.

1 Introduction

The use of phrases as part of a text representation or indexing language has been investigated since the early days of information retrieval research. Cleverdon, for example, included phrase-based indexing in the Cranfield studies (1966). Salton (1968) also described a variety of experiments using phrases in the SMART system. Certainly, there has always been the feeling that phrases, if used correctly, should improve the specificity of the indexing language and, consequently, the quality of the text representation. The experimental results obtained with phrases do not, however, support this intuition. These results have been very mixed, ranging from small improvements in some collections to decreases in effectiveness in others[1].

Fagan's recent thesis (1987) is one of the most comprehensive studies of automatic indexing using phrases, in that he used both "statistical" and "syntactic" phrases and varied a number of factors in the phrase formation process. A statistical phrase is defined by constraints on the number of occurrences and co-occurrences of its component words and/or the proximity between occurrences of components in a document. A syntactic phrase may be characterized by some of the same criteria as a statistical phrase, but in addition must obey some constraint on the syntactic relationships among its component words. Fagan's results showed significant increases in some collections with statistical phrases, but none using syntactic phrases. It should also be pointed out that his improvement figures obtained with collections such as CACM[2] were relative to quite low baseline results. Improvements over the best single word baselines might have been considerably smaller. In experiments with user-identified phrases using both Fagan's algorithm and a probabilistic algorithm, we found that neither provided results significantly different from single word representations (Croft and Das, 1990).

Despite the significant amount of work on phrases, we feel that the relationship of phrases to the retrieval model has not been sufficiently examined. For example, should a phrase be treated as an index term, similar to index terms derived from single words, or should it be treated as a relationship between index terms? The answers to questions such as these are not obvious and have significant implications for retrieval algorithms. One of the goals of this paper is to clarify the issues involved in using phrases with a retrieval model.

In commercial systems, searchers express linguistic structure (e.g. phrases) using Boolean expressions containing operators such as AND ($\wedge$), OR ($\vee$), word-level

[*] Current address: West Publishing Company, St. Paul, Minnesota.

[†] Current address: Center for Information and Language Studies, University of Chicago, Chicago, Illinois.

[1] We assume retrieval effectiveness is measured in terms of recall and precision.

[2] A test collection consists of a set of documents, a set of queries, and lists of the relevant documents for each query. The Communications of the ACM (CACM) collection is described in section 4.1.

proximity, sentence-level proximity, and paragraph-level proximity. The concept *information retrieval*, for example, may be expressed by (*information* $\wedge$ *retrieval*), or by using a proximity operator such as (*information* within 3 words of *retrieval*). Structure in the query is used to describe how the phrase, or other linguistic construct, can be detected in a document text. In previous work, we have used Boolean queries to identify potential phrases that were used in a probabilistic model incorporating term dependency (Croft, 1986). In that work, phrases were interpreted as specifying term dependencies.

In this paper, we take a different approach. Phrases identified in a natural language query are used to construct a structured query, which is then used in a probabilistic model based on inference nets (Turtle and Croft, 1991). This represents a step towards our overall research goal, which is to build a complex, inference net-based representation of an information need through natural language analysis and user interaction.

In the following section, we review previous work. We start by describing the inference net model which is the basis of our experiments. We then describe research on phrases, emphasizing the different ways phrases have been treated in retrieval models. Instantiating each of these models in the form of an inference network enables the similarities and differences among them to be clearly seen. The last subsection reviews work that uses Boolean queries and operators such as proximity in statistical retrieval models.

In section 3, we give an overview of our approach to building structured queries, and describe the specific techniques used for phrases in this paper. Section 4 presents the experimental results and a discussion of those results. Finally, in section 5, we indicate future directions and discuss the importance of large document collections.

2 Previous Work

2.1 The Inference Net Model

The inference net model (Turtle and Croft, 1991) is used as the basis for the comparisons of different treatments of phrases, and for the experiments in section 4. It is a probabilistic retrieval model in that it follows the probability ranking principle (Robertson, 1977). Typically, a probabilistic model calculates P(Relevant|Document,Query), which is the probability that a user decides a document is relevant given a particular document and query (Fuhr, 1989). The inference net model takes a slightly different approach in that it computes P(I|Document), which is the probability that a user's information need is satisfied given a particular document. More specifically, we consider an information need as a complex proposition about the content of a document, with possible values true and false. Queries are regarded as representations of the information need.

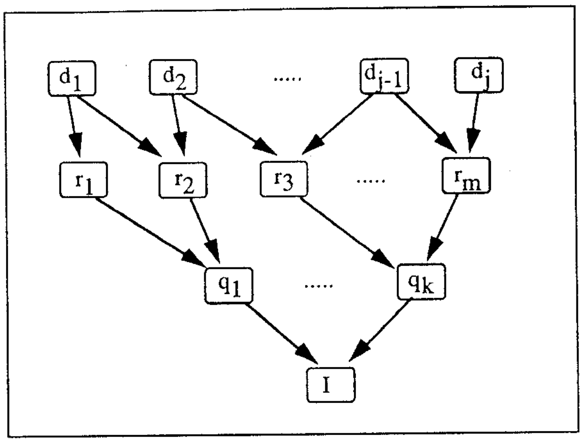

Figure 1: Basic Inference Network: d_i's are document nodes, r_i's are concept nodes, q_i's are query nodes, and I represents the user's information need.

The major difference between the inference net model and other probabilistic models is that it emphasizes the use of multiple sources of evidence to calculate P(I|Document). Different representations of the document content, different representations of the information need, and domain knowledge such as a thesaurus can all be taken into account under this model. For this paper, the major advantages are that structured queries have a natural interpretation in the inference net model and different forms of the model can be shown using a diagram. These features of the model are discussed below.

A Bayesian inference network (Pearl, 1989) is a directed, acyclic dependency graph (DAG) in which nodes represent propositional variables or constants and edges represent dependence relations between propositions. If a proposition represented by a node p "causes" or implies the proposition represented by node q, we draw a directed edge from p to q. The node q contains a matrix (a *link* matrix) that specifies $P(q|p)$ for all possible values of the two variables. When a node has multiple parents, the matrix specifies the dependence of that node on the set of parents and characterizes the dependence relationship between that node and all nodes representing its potential causes. Given a set of prior probabilities for the roots of the DAG, these networks can be used to compute the probability or degree of belief associated with all remaining nodes.

Figure 1 shows the basic inference network used in this paper. It consists of a document network and a

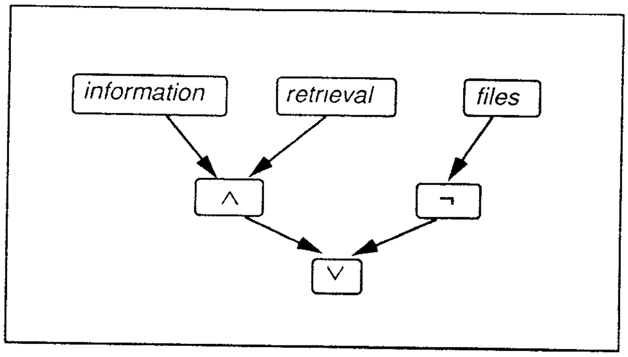

Figure 2: Structured query network for the query $(information \wedge retrieval) \vee \neg files$

query network. The document network is built once for a collection and its structure does not change during query processing. The query network consists of a single node which represents the user's information need and one or more query representations which express that information need. A query network is built for each information need and is modified through interactive query formulation or relevance feedback.

The document network consists of document nodes (d_i's) and concept representation nodes (r_k's). Each document node represents an actual document in the collection and corresponds to the event that a specific document (i.e. text content in this simple model) has been observed. We represent the assignment of a specific representation concept to a document by a directed arc to the representation node from each node representing a document to which the concept has been assigned. A representation node contains a specification of the conditional probability associated with the node given its set of parent document nodes.

The query network is an "inverted" DAG with a single leaf that corresponds to the event that an information need is met and multiple roots that correspond to the concepts that express the information need. A set of intermediate query nodes may be used to describe complex query networks such as those formed with Boolean expressions. Figure 2 shows the query network for the query $(information \wedge retrieval) \vee \neg files$. Each of the Boolean operators has a corresponding canonical link matrix form (Turtle and Croft, 1991). Turtle (1990) showed that this inference network model of structured queries is at least as effective as the "extended Boolean" version of the vector space model (Salton, Fox and Wu, 1983).

For retrieval, a query network is built through interaction with the user, and attached to the document network. This allows us to compute the probability that the information need is met for any particular document and, consequently, to produce a ranked list of documents.

2.2 Phrases

The use of phrases in experimental IR systems can be discussed in terms of the following issues:

1. What evidence is used to determine if a phrase exists in a document or query?
2. Are phrases separate concepts or are they relationships between concepts?
3. What is an appropriate weighting for phrases?
4. Should the use of phrases affect which single word terms are used for indexing queries and documents?
5. Are phrases part of the document indexing or only identified during search?

The first issue is the basis for distinguishing between statistical phrases and syntactic phrases. As mentioned before, syntactic phrase indexing techniques use linguistic evidence to identify phrases. Both template-based and parser-based techniques have been used. Dillon and Gray's FASIT system (1983) is typical of the template-based approaches, where the syntactic categories of words in documents are identified and patterns of adjacent categories are matched against a library of templates (such as `<adjective noun>`). Parser-based approaches have used parsing techniques and grammars of varying sophistication to analyze the document and query text. For example, Fagan (1987) used the PLNLP parser to produce a complete parse of the document text, whereas Smeaton (1988) used a simpler grammar. In both cases, specific linguistic constructs (e.g. noun phrases) in the parse tree are then identified as phrases for indexing. It is also possible to use semantic evidence to refine the phrase extraction. Sparck Jones and Tait (1984), for example, used a syntactic parser together with general semantic information to analyze queries.

Statistical phrase indexing techniques, on the other hand, use information about the co-occurrence of words to identify phrases. It is possible, for example, to identify pairs of words that are strongly associated using measures such as the expected mutual information measure (Van Rijsbergen, 1979). Of course, two words may tend to co-occur for other reasons than being part of the same phrasal concept. For instance, the hypothesis that synonymous or nearly synonymous words will be used together in documents has been the basis of considerable research on term clustering. If linguistic or statistical clues can be used to distinguish these two types of co-occurrences, and possibly others as well, then the choice of a phrase model or thesaurus relationship can be made on a case-by-case basis (Lewis, 1991; Krovetz and Croft, 1991).

The statistical phrase indexing procedure used by Fagan (1987) is an extension of that used by Salton, Yang and Yu (1975). The basic algorithm involves selecting pairs of words from document and query texts, where the individual words and the form of their co-occurrence satisfy various criteria. Fagan found that the information about term specificity and relationships among words in text that is provided by document frequency, proximity, and frequency of co-occurrence did not improve phrase selection. For example, the best effectiveness improvements obtained in his experiments with the CACM collection (see section 4.1) were with "phrases" formed from *every* pair of words in the documents and queries. The only restriction was that pairs occurring more than 90 times in the collection were rejected.

Another form of evidence that has been used in previous experiments is user judgments (Croft, 1986; Croft and Das, 1990). In these experiments, users were asked to identify phrases (and words) in initial query statements and relevant documents. This is potentially a very accurate form of evidence, although it places a heavy burden on the interface designer and, to some extent, the users of the system. User input during query formulation and relevance feedback has been shown to be generally effective (Harman, 1988; Croft and Das, 1990), but results with phrase identification have been mixed.

Issues 2 through 4 are best discussed by referring to the inference net models in Figure 3. These inference nets show alternative ways of modeling phrases in a probabilistic retrieval model, and can also be used to describe the use of phrases in the vector space model, for example. In these small networks, r_i, r_j are representation concepts corresponding to two words in the text of the document d_m. The phrase p_k is also a representation concept that corresponds to a phrase in the text consisting of the two words. Q represents a query. As an example, r_i and r_j may correspond to the occurrence of *information* and *retrieval*, respectively, and p_k would then correspond to occurrence of the phrase *information retrieval.*

In the first model (Figure 3(a)), a phrase is treated as a separate representation concept, independent of the concepts corresponding to the component words. The belief in the phrase concept can be estimated using evidence about the component words and the relationship between them, including linguistic relationships. The presence of a query phrase concept in a document will increase the probability that the document satisfies the query (or information need). This is the model used in Smeaton's work (1986), where all phrases had the

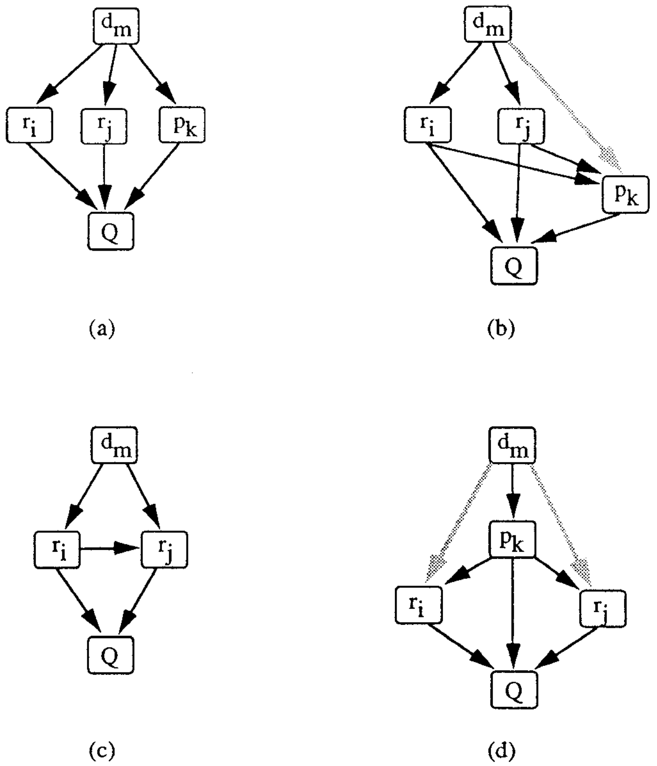

Figure 3: Alternative Phrase Models:
(a) Belief in phrase independent of belief in components
(b) Belief in phrase dependent on belief in components
(c) Phrase is a dependency relationship between components
(d) Belief in components dependent on belief in phrase

same belief (or weight), and will also be used in the experiments described in section 4.

The second model (Figure 3(b)) shows the case where the belief in the phrase concept depends on the belief in the concepts corresponding to the component words. This is the model used by Fagan (1987) for both statistical and syntactic phrases. Fagan calculates the weight associated with the phrase as the average of the "tf.idf" weights of the component words. A tf.idf weight is formed using a combination of the relative frequency of a word in a document (tf) and the inverse of the relative frequency of the word in the document collection (idf). Specific forms of this weight are discussed by Salton and McGill (1983) and Turtle (1990). In Fagan's work, the score for a matching phrase was added to the document score from the individual words. This model is also used in our experiments with structured queries, where phrases are represented as the AND of two concepts (similar to *information retrieval*in Figure 2). The gray arrow in Figure 3(b) indicates that the belief in the phrase concept may depend partially on evidence from the document text. For instance, with Fagan's syntactic phrases, specific syntactic relationships had to be observed in the text for a phrase to be present.

The third model (Figure 3(c)) is a term dependence model. Here the phrase is not represented as a separate concept, but as a dependence between the concepts corresponding to the component words. A document that contains both words will be more likely to satisfy the query due to the increased belief coming from r_i to r_j. This model was used by Croft (1986) in experiments with natural language and Boolean queries. We now believe that this model is less appropriate for phrases than for capturing other relationships between concepts, such as those in a thesaurus. In this case, the important behavior is that a document that contains only concept r_i will also generate a significant belief in r_j due to this dependence.

The final model (Figure 3(d)) has not been used in previous work, but has some justification. In this model, the belief in the phrase concept is established using evidence from the document text, and the beliefs in the concepts representing component words are derived from the belief in the phrase and, to a lesser extent, the document text. This model makes explicit the conditional dependence between the component concepts, and also the idea that all component words of a phrase might not be used in the text representation. For example, if the phrase *prime number* occurs in a document, should both *prime* and *number* also be used in the representation? We address this issue in the experiments in section 4. This model has the disadvantage that the belief in the component word concepts is limited by the belief in the phrase concept.

Each of these is a formal model of phrase indexing and retrieval. The models do not, however, specify how the beliefs (or weights) of phrases should be estimated. In this paper, we will investigate some alternative weighting schemes in the context of models (a) and (b). Fuhr and Buckley (1990) suggest a technique that could be used to learn the appropriate phrase weight, but we will not address this further in this paper. One of the major problems with estimating weights for phrases is the limited size of test collections. In these collections, most phrases occur infrequently and belief estimates are consequently inaccurate. Although the experiments reported in section 4 use the relatively small CACM collection, we are currently working with a larger collection.

The final issue is whether phrases are part of document and query indexing, or just query indexing. This is essentially an implementation decision, in that any of the phrase models described can be carried out by identifying phrases in the query and then scanning documents for occurrences of those phrases at query time. The difference between query and document representation concepts is, in this case, very little. Some phrase indexing methods may also be impractical for large collections. For example, Fagan's technique of using virtually every pair of words in a document as an indexing phrase would result in unreasonable storage overheads if an inverted file of index terms was used. If phrase indexing is not done prior to search, however, it will be necessary for the file organization for the documents to contain sufficient information to identify and weight phrases. This may involve, for example, storing word position information or providing access to the full text.

2.3 Structured Queries

Although Boolean query languages may be difficult for people to use, there is considerable evidence that trained searchers can achieve good search effectiveness using them. It appears that a structured query containing operators such as AND, OR, and proximity can be used to describe accurate representations of information needs. As an example, consider that in experiments with the extended Boolean model (Salton, Fox and Wu, 1983) and the network model (Turtle and Croft, 1991), structured queries were more effective than simpler queries consisting of a set of weighted terms.

When a user presents an information need in the form of a natural language query, they specify, by their choice of particular linguistic relationships, a variety of meaningful connections between the words in the query. Treating such a query as a set of weighted terms ignores these connections. One of the advantages of a structured query may be to capture, in a form easily processed by computer, some of the relational structure normally expressed in natural language.

There has been some preliminary research on the best translation into Boolean operators (AND and OR) of linguistic relationships from queries. Croft (1986) compared the use of Boolean and natural language queries as sources of word pairs for the same term dependency model. Das-Gupta (1987) proposed an algorithm, using both syntactic and semantic information, for deciding when the natural language conjunction *and* should be interpreted as a Boolean AND and when as a Boolean OR. Preliminary comparisons showed a high degree of agreement with the translations of human experts, though the results must be considered tentative, due to a variety of experimental limitations discussed by Das-Gupta.

Smith (1990) presented a complex algorithm for translating a full syntactic parse of a natural language query into Boolean form. She compared the resulting Boolean queries with both manually and statistically produced ones on three test collections using a variety of p-norm interpretations for Boolean operators. The syntactically produced Booleans performed substantially better than statistically produced Booleans and comparably to manually produced ones.

Smith's work strongly suggests that some of the effectiveness of structured queries comes from capturing relationships which are directly reflected in the syntactic structure of a natural language query. However, the complexity of Smith's algorithm, including its use of ad hoc lists of words to be rejected from queries, makes it difficult to draw conclusions about the relative effectiveness of Boolean operators for capturing different linguistic relationships.

To understand more about the connection between the linguistic relationships in natural language queries and the Boolean operators used in structured queries, we compared natural language queries from the CACM collection with the Boolean queries that Cornell graduate students derived from them (Smith, 1990). We found that 82% of the conjunctions (ANDs) of words and 50% of the disjunctions (ORs) of words correspond to relatively simple linguistic structures in the natural language queries. Indeed, 61% of the ANDed groups of words correspond to direct modification relationships between the heads of noun phrases, or between an adjective and the noun it modifies. This suggests that reasonable Boolean queries might be produced by translating certain syntactic relationships into Boolean operators.

The above studies all focused on the connections between linguistic relationships and Boolean operators. Proximity phrases provide another means for structured queries to capture linguistic relationships. As an example of how proximity may help, consider the phrase *operating system*, which occurs several times in CACM queries. Of the 64 instances of the two words *operating* and *system* occurring within 3 words of each other in documents, only 3 were not referring to the concept *operating system*. On the other hand, all 27 instances of these two words occurring at greater distances apart were in documents that were not about the concept *operating system*.

An example of the use of a proximity interpretation for phrases is the RUBRIC system (Tong and Shapiro, 1985), where simple rules based on co-occurrence and proximity of words in text are used to infer the presence of query concepts. Gay and Croft (1990) describe a study of nominal compounds (noun groups) in CACM queries. They found that the concepts corresponding to these nominal compounds could be very accurately identified in document texts using simple proximity. Fagan's results (1987), on the other hand, showed that interpreting statistically produced phrases as requiring close document proximity of words did not improve effectiveness over allowing unlimited proximity.

In section 4, we describe an approach to using proximity in structured queries. As with phrase weighting, the use of proximity is difficult to evaluate with the CACM collection. In this case, the problem is the small size of the CACM documents, which contain only abstracts and an average of only 20 unique stems per document. In these small documents, unlimited proximity or co-occurrence at the document level will be equivalent to restricted proximity (e.g. same paragraph) in longer, full-text documents. In future experiments with collections of full-text documents we will in particular investigate methods for translation of phrases and linguistic relationships into ANDs, ORs, and proximity phrases on a case by case basis, rather than imposing a single model on all phrases.

3 Building Structured Queries from Natural Language

In order to obtain accurate descriptions of information needs while avoiding the problems of complex query languages, our research goal is to build structured queries by a combining natural language analysis of free text user input with an interface that aids user specification of query structure (e.g. Anick, 1990). The structured queries will be represented as inference networks and will contain information about concepts that represent the information need, their relative importance, and relationships between them (Croft and Das, 1990).

In this paper, we address one part of this research goal, which is to evaluate structured queries containing phrases. Specifically, we report the results of experiments designed to test the following hypotheses:

> *Hypothesis 1*: Structured queries incorporating phrases will be more effective than unstructured queries.
> *Hypothesis 2*: Phrases selected automatically

	Number of Phrases (% manual) - 50 queries						
	manually selected	parsed		tagged		tagged + dict	
No filtering	197	407	(71%)	148	(69%)	191	(89%)
Corpus filtering	151	221	(50%)	119	(52%)	159	(72%)

Table 1: Numbers of phrases generated using various techniques

will perform as well as phrases selected manually.

In the next section, we describe experiments using a variety of phrase models and belief estimates. Two methods for deriving syntactic indexing phrases from natural language queries are investigated. These are a parser-based syntactic phrase extraction procedure described by Lewis and Croft (1990), and the stochastic tagging system developed by Church (1988). These phrases are compared to phrases selected manually from the queries.

4 The Experiments

Two sets of experiments were conducted to test our hypotheses. The first set (Section 4.1) was intended to test methods for representing phrase information in queries and to test whether these approaches improve retrieval performance. These results bear directly on Hypothesis 1.

The second set of experiments (Section 4.2) was designed to test Hypothesis 2. These tests compare the performance of different methods for automatically selecting phrases with that obtained using manually selected phrases.

All experiments were done using the CACM test collection (Salton, Fox and Wong, 1983). Our version of this collection contains 3204 abstracts from *Communications of the ACM*, along with 50 queries in both natural language and Boolean form. One set of queries was formed by taking the natural language queries and having a M.S. computer science student identify important phrases in the text. This query set is the basis of the manually selected phrase experiments. For each query, a list of relevant documents is provided. Standard recall-precision tables (Salton and McGill, 1983) are used to evaluate and compare the retrieval results.

Table 1 shows the numbers of phrases produced by the various techniques used in this paper. This table will be referred to in the following sections. The figures in parentheses show the percentage of the manually selected phrases that were contained in the various sets of phrases.

4.1 Belief estimates for phrases

The occurrence of a phrase in a document represents evidence supporting the assignment of a representation concept to a document. In this sense, a phrase is similar to a normal term, but the estimation techniques used for normal terms must be extended to accommodate phrases.

	Precision – 50 queries		
Recall	NL	*and*-based	
10	67.6	68.5	(+1.2)
20	54.3	61.8	(+13.8)
30	48.7	53.4	(+9.7)
40	42.5	43.8	(+3.0)
50	35.8	37.4	(+4.6)
60	28.3	29.1	(+2.7)
70	19.7	22.3	(+13.2)
80	15.7	18.1	(+15.6)
90	10.6	12.7	(+19.6)
100	8.0	8.9	(+11.8)
average	33.1	35.6	(+7.5)

Table 2: Performance of manually constructed queries – *and*-based phrases

Three methods for estimating belief in phrases were tested:

1. treating a phrase as a conjunction of single terms (term co-occurrence in a document),
2. treating a phrase as a proximity relation, and
3. a hybrid method in which belief depends on the frequency of the individual terms as well as the frequency of the phrase.

The first method is based on the phrase model shown in Figure 3(b), whereas the other two are based on the model in Figure 3(a).

4.1.1 Conjunctive phrases

The first approach, reported in Turtle (1990), models a phrase as a co-occurrence of the component terms in a document; a query is formed by *and*ing the component terms for each phrase and combining the phrasal subexpression with any remaining terms using a probabilistic *sum* operator. In an inference network a two-term *and* is modeled as the product of the beliefs for the individual terms. Since beliefs lie in the range [0..1], the belief assigned to a phrase will be lower than that assigned to either component term. The probabilistic sum operator computes the mean of the beliefs assigned to the component terms. Using this *and*-based model with queries

	Precision (% change) - 50 queries				
Recall	NL	*and*-based		proximity	
10	67.6	68.5	(+1.2)	65.9	(−2.6)
20	54.3	61.8	(+13.8)	56.2	(+3.6)
30	48.7	53.4	(+9.7)	50.0	(+2.7)
40	42.5	43.8	(+3.0)	39.5	(−7.1)
50	35.8	37.4	(+4.6)	33.9	(−5.2)
60	28.3	29.1	(+2.7)	28.3	(−0.2)
70	19.7	22.3	(+13.2)	21.4	(+8.6)
80	15.7	18.1	(+15.6)	16.1	(+2.5)
90	10.6	12.7	(+19.6)	10.3	(−3.0)
100	8.0	8.9	(+11.8)	7.0	(−12.3)
average	33.1	35.6	(+7.5)	32.9	(−0.8)
top 10	35.2	38.0	(+8.0)	37.2	(+5.7)

Table 3: Performance of manually constructed queries - proximity-based phrases

	Precision (% change) - 50 queries				
Recall	*and*	hybrid		Fagan	
10	68.5	68.5	(+0.0)	69.2	(+1.1)
20	61.8	57.4	(−7.1)	62.2	(+0.8)
30	53.4	51.0	(−4.5)	53.0	(−0.7)
40	43.8	44.3	(+1.2)	43.6	(−0.3)
50	37.4	37.6	(+0.4)	37.4	(−0.0)
60	29.1	33.5	(+15.3)	29.6	(+1.8)
70	22.3	24.5	(+10.1)	22.4	(+0.3)
80	18.1	19.9	(+9.6)	18.2	(+0.2)
90	12.7	13.2	(+3.8)	12.7	(−0.2)
100	8.9	10.0	(+12.3)	9.0	(+0.8)
average	35.6	36.0	(+1.1)	35.7	(+0.4)

Table 4: Comparison of *and*-based, hybrid, and Fagan's belief estimates

in which phrases and single terms were identified manually resulted in significantly better performance than with the original natural language queries (Table 2).

4.1.2 Proximity phrases

In the second model, a phrase is viewed as a sequence of terms occurring in close proximity in a document. In the *and*-based model, a document containing any term from a phrase is assigned a belief greater than the default belief. This belief increases with the number of terms in common with the phrase, but is based only on the beliefs associated with the single terms. In the proximity model, only those documents containing all of the terms in a phrase in the required proximity are assigned a belief greater than the default. This belief estimate depends on the number of documents in which the proximity relation is satisfied and is independent of the beliefs associated with the single terms. In the proximity model, a phrase is viewed as an independent representation concept whose belief is based on the within document frequency (tf) and the inverse document frequency (idf) of the phrasal unit. See Turtle (1990) for more details on the use of tf and idf to estimate belief for concepts in inference nets. The width of the proximity window is set to three for these experiments, based on the results of Gay and Croft (1990).

As shown in Table 3, the proximity model performs significantly worse than the conjunctive phrase model and performs about as well as the original natural language query. Performance with this model suffers because the proximity model is too "strict" - documents that contain only one of the terms in the phrase or do not satisfy the proximity constraints are assigned the same belief as a document containing none of the terms. The poor performance of the proximity model is principally due to not recognizing single term matches rather than to ignoring documents that do not satisfy proximity constraints. Relaxing the proximity window produces very few additional matches.

The poor performance of the proximity model is also due, in part, to the composition of the CACM collection. The records in this collection are short (many documents have no abstract and abstracts that are present generally consist of a single paragraph) and the collection only contains 3204 records. Many legitimate query phrases occur infrequently, if at all. The use of proximity is a precision enhancing mechanism which sacrifices recall. As such, it is not particularly effective with collections of short documents. The use of an *and*-based estimate is a recall enhancing device. We would expect the proximity-based estimate to be more effective than an *and*-based estimate for collections containing large documents. This view is supported by recent experiments (Harman and Candela, 1990). Indeed, if we compare raw precision in the top ten documents retrieved from the CACM collection, queries using proximity phrases perform significantly better than the original natural language queries and nearly as well as the *and*-based phrases.

4.1.3 Hybrid approaches

To test the hypothesis that proximity-based phrases enhance precision while *and*-based phrases enhance recall, we tested a series of hybrid phrase models that attempt to combine the best features of both. All of these hybrids estimate belief based on the proximity phrase if it is present in a document and use some other estimate based on the single term belief if the proximity constraints are not met. The estimates based on single-term beliefs that were tested include the original *and*-based estimate (the product of the beliefs), the mean of the beliefs, and the maximum of the single term beliefs. Of these, the best hybrid phrase operator used the maximum of the single term beliefs for a document if the proximity constraints were not met. The performance of the hybrid phrase estimate is not significantly better than the original *and*-based formulation on the CACM collection (Table 4). However, initial work with a collection containing larger documents suggests that the hybrid estimate can improve average precision by 10-20% over the *and*-based estimate.

To test the relative importance of the phrases and single terms, tests were conducted using manually identified phrases and terms, manually identified phrases with no single terms, and manually identified phrases with all single terms from the original queries. As shown in Table 5, dropping the single terms significantly degrades retrieval performance. Adding all single terms performs about as well as using only manually selected terms.

4.1.4 Comparison with earlier phrase model results

To compare our work with Fagan (1987), we implemented a phrase model in which the belief estimate for a phrase is simply the mean of the belief estimates for the component terms. With this model, we use essentially the same method to combine beliefs for terms in a phrase as we would use to combine beliefs for terms in a query. By computing the phrase weight in this way we are, in effect, normalizing the phrase weight so that it acts as a single term when combined with other terms in the query.

As shown in Table 4, there is little difference between the *and*-based, hybrid, or Fagan estimates on the CACM collection. Again, initial work with larger collections suggests that the hybrid estimate will perform better than either the *and*-based or Fagan's estimate on collections of large documents. We should also point out that the average precision figures obtained using phrases with inference nets are significantly higher than those reported in Fagan (1987). Specifically, an average of .36 compared to an average of .32.

4.2 Automatic versus manually selected phrases

The results of the last section show that incorporating information about manually selected phrases can improve retrieval performance. While a user could provide this kind of information during an interactive session, we are of course interested in techniques that could identify useful phrases automatically, sparing the user effort. In this section we describe experiments with three methods for automatically recognizing phrases: a parser based primarily on phrase syntax (Section 4.2.2), a stochastic phrase bracketer which incorporates part of speech information obtained from a training corpus (Section 4.2.3), and phrases obtained from a dictionary (Section 4.2.4).

In addition to the phrase recognition method, three other variables were considered: whether corpus filtering (Section 4.2.1) is used, what belief estimate is used (*and*-based, proximity, or hybrid), and the method used to select single terms (no single terms, all single terms from the original queries, or some subset). We will focus on recognition technique and corpus filtering in the following sections. Results for the remaining two variables are relatively independent of these and will be summarized here.

The performance of the belief estimates described in the last section for manually selected phrases are independent of the remaining variables. In general, the *and*-based and the hybrid estimates behave similarly and both perform significantly better than the proximity-based estimate when used on the CACM collection. Again, we hypothesize that the performance of the hybrid and proximity-based estimates will improve for collections of large documents and that *and*-based performance will degrade.

The method used to select a set of single terms to be included with phrases has a major impact on retrieval performance. In general, eliminating single terms significantly degrades retrieval performance since many of the concepts contained in the original query are not described by phrases. It is also clear that not all of the single terms from the original query are required for effective retrieval. The manually constructed queries eliminate 60% of the single terms from the original query, but the performance of these queries is essentially equivalent to that obtained with queries that contain all single terms (in addition to the phrases). Strategies for selecting the set of single terms to be included with phrases were examined, but these strategies performed about as well as simply using all single terms from the original query. One factor that should probably influence the strategy used to select single terms is phrase quality. For high-quality phrases, such as those found in dictionaries and with high frequency in the collection, the component terms should be dropped. For phrases that are less likely to be correct, the component terms should be retained. Unless otherwise noted, all results in the remainder of this section use all single terms from the original query in addition to the set of selected phrases.

4.2.1 Corpus filtering

The quality of automatically selected phrases varies considerably depending upon the technique used to generate the phrases and on the quality of the original natural language query. In many cases it may be useful to apply some form of test to generated phrases in an attempt to screen out low quality ones. One technique that can be used to eliminate spurious phrases is corpus phrase filtering (Fagan, 1987) in which candidate phrases are retained only if their collection frequency exceeds some threshold (in our case, more than one

	Precision (% change) – 50 queries		
	man. phrases	man. phrases	man. phrases
Recall	man. terms	no terms	all terms
10	68.5	65.2 (−4.8)	68.5 (+0.0)
20	57.4	54.1 (−5.7)	58.6 (+2.1)
30	51.0	45.7 (−10.5)	51.0 (−0.2)
40	44.3	39.7 (−10.3)	43.7 (−1.5)
50	37.6	33.6 (−10.6)	38.4 (+2.2)
60	33.5	28.4 (−15.3)	31.9 (−4.9)
70	24.5	20.2 (−17.8)	23.7 (−3.4)
80	19.9	15.4 (−22.4)	18.0 (−9.6)
90	13.2	11.2 (−14.8)	12.8 (−3.2)
100	10.0	9.3 (−7.5)	10.2 (+1.4)
average	36.0	32.3 (−10.3)	35.7 (−0.9)

Table 5: Performance effect of single term selection

	Precision – 50 queries	
Recall	unfiltered	filtered
10	68.5	68.2 (−0.5)
20	58.6	58.0 (−1.0)
30	51.0	50.1 (−1.8)
40	43.7	43.0 (−1.6)
50	38.4	37.1 (−3.5)
60	31.9	30.8 (−3.3)
70	23.7	22.9 (−3.6)
80	18.0	17.7 (−1.8)
90	12.8	11.9 (−6.8)
100	10.2	9.3 (−8.8)
average	35.7	34.9 (−2.2)

Table 6: Effect of corpus phrase filtering with manually constructed queries

occurrence). Table 1 shows that corpus filtering does reduce the number of phrases for each technique used, but it is also clear that a number of reasonable phrases are eliminated (using manually selected phrases as a guideline).

The effectiveness of corpus phrase filtering depends heavily on the quality of the original phrases. For manually selected phrases, eliminating query phrases that do not occur more than once in the collection actually hurts performance slightly[3] (Table 6). Since a user deliberately produced the phrase, there is strong reason to treat it as high quality even if it does not appear in the CACM collection. Some of these user-produced phrases were not commonly used terminology during the period covered by the CACM collection (e.g., *window manager*, *horizontal microcode*) while others simply do not occur in the small text sample (e.g., *command interpreter*, *complexity class*, *type compatibility*). The individual words comprising these phrases tend to be good content descriptors and the inclusion of the phrases tends to help overall performance.

Automatically selected phrases tend not to be as reliable as manually selected phrases and corpus phrase filtering improves performance for these phrases. For consistency, results will use corpus phrase filtering unless otherwise noted. This will mean that manually selected phrase results are somewhat understated since better performance can be achieved without corpus phrase filtering.

Work with larger collections suggests that corpus term filtering can be effective. With this technique, words from phrases with very high collection frequencies are removed from the set of single word terms included in the query. Essentially, very high frequency phrases (e.g., *operating system*, *computer system*) are assumed to match the query only when they occur as a phrase in a document and documents receive no credit for single term occurrences. Again, this technique gives only slight performance gains with the CACM collection.

4.2.2 Syntactic phrases from a partial parse

One method of automatically generating phrases is to parse the query or document text, and extract all pairs of words which occur in specified syntactic relationships in the parser output. For this experiment, we used a syntactic phrase generation system which attempts to generate all pairs of non-function words that are heads of syntactic structures connected by a grammatical relationship (Lewis and Croft, 1990). Examples are a verb and the head noun of a noun phrase which is its subject, and a noun and a modifying adjective. The system analyzes only those sentence constituents below the clause level, relying a set of heuristics to produce phrases from adjacent constituents. The lexicon used is the Longman Dictionary of Contemporary English (LDOCE) (Boguraev and Briscoe, 1987) which provides

[3] When using strict proximity, performance is unaffected since all documents are assigned the same belief for the phrase

syntactic categories for about 35,000 words. A simple analyzer for inflectional morphology augments this vocabulary considerably.

As shown in Table 7, queries using the syntactic phrases produced by the system described above perform significantly worse than either the natural language or manually constructed phrases. The syntactic parser produces a very large number of candidate phrases (Table 1), many of which do not describe query content. While many of these noise phrases are eliminated by corpus phrase filtering, phrases of questionable value remain (e.g., *implementations work*, *algorithms programs*, *separation corresponding*) which degrade performance. Since the set of syntactic phrases is large and contains most of the useful phrases from the original queries, it is possible that a better filtering technique could be used to significantly improve the performance of syntactic phrases. More accurate parsing, using a more comprehensive grammar, might also help.

4.2.3 Syntactic phrases from a stochastic tagging

The stochastic tagger developed by Church (1988) assigns parts of speech to words based on knowledge of lexical and contextual probabilities of occurrence. In addition, the boundaries of simple noun phrases are identified.

We investigated using exactly the simple noun phrases tagged by the Church tagger as syntactic indexing phrases. The number of phrases produced by the Church tagger is considerably lower than the number produced by the syntactic parser (Table 1), but a much higher proportion of these phrases are reasonable, based on comparison with the manually selected phrases. The lower number of phrases produced by the tagger system results both from its lower error rate, and because it generates indexing phrases from fewer linguistic structures than the parser-based system. Which of these factors is the bigger contributor to improved performance remains to be established.

Queries using tagged phrases perform slightly better than the original natural language queries and somewhat worse than the manually selected phrases (Table 7, but recall that manual phrases perform slightly better without corpus phrase filtering).

4.2.4 Phrases from a dictionary or thesaurus

Another possibility for identifying phrases in queries is to use a machine-readable dictionary or domain-specific lexicon. For these experiments, we used a very general source of phrases for computer science - the ACM Computing Review Classification System (1987 version). Because there are very few phrases in this source, we used those identified in queries as additions to the tagged phrases. A dictionary phrase was identified if the exact form of the phrase occurred in the query. Table 1 shows that, although the dictionary only added a small number of phrases, almost all of them were present in the manually selected set.

When the dictionary phrases are added to the tagged phrases (Table 8), performance improves slightly. Using corpus term filtering, we removed single words from queries when they were contained in phrases that occurred in more than 50 documents. Performance with these abbreviated queries improved further.

4.2.5 Summary

Combining tagged and dictionary phrases results in queries that perform better than the original natural language queries, although not as well overall as queries containing manually selected phrases and content bearing terms (Table 9). The automatically produced phrases did outperform the manually selected phrases at high precision levels, which may be important from a user's perspective. It is likely that these performance levels can be further improved with better strategies for selecting phrases and for selecting the set of single terms to be included in the query.

As mentioned earlier, initial work suggests that the importance of representing structure in queries increases with document and collection size. Proximity appears to be an effective tool for identifying phrases in documents and proximity-based and hybrid schemes appear to work substantially better than the simple term co-occurrence techniques investigated thus far.

5 Conclusion

Based on the results in section 4, we can accept the hypothesis that phrases (and structured queries) improve retrieval effectiveness. The results also support the second hypothesis in that there is little difference between the effectiveness of manually and automatically selected phrases. As mentioned previously, we are currently repeating these experiments with a much larger corpus. We, and others, are observing that, on larger collections, proximity becomes a much more useful source of evidence about phrases, and phrases contribute more to effective retrieval.

Our future research will concentrate on improving the techniques for building structured queries. We are looking at other types of information to include in the query, such as relative importance of query concepts and concepts related to those in the original query. We are also studying methods for capturing relationships between concepts that go beyond simple linguistic structures such as phrases. For example, in Boolean queries, experts often form the AND of two concepts which are not phrase components. This implies a strong rela-

	Precision (% change) – 50 queries						
Recall	NL	manual		parsed		tagged	
10	67.6	68.2	(+0.8)	59.2	(−12.5)	68.4	(+1.2)
20	54.3	58.0	(+6.9)	51.4	(−5.3)	57.3	(+5.6)
30	48.7	50.1	(+2.8)	41.5	(−14.7)	49.4	(+1.5)
40	42.5	43.0	(+1.1)	36.4	(−14.4)	41.4	(−2.5)
50	35.8	37.1	(+3.7)	30.5	(−14.8)	35.4	(−1.0)
60	28.3	30.8	(+8.8)	26.1	(−7.9)	29.0	(+2.3)
70	19.7	22.9	(+16.1)	19.7	(+0.3)	21.3	(+8.3)
80	15.7	17.7	(+12.5)	15.6	(−0.7)	17.0	(+8.4)
90	10.6	11.9	(+12.0)	10.1	(−4.5)	11.3	(+6.8)
100	8.0	9.3	(+16.1)	7.2	(−9.4)	8.5	(+6.0)
average	33.1	34.9	(+5.3)	29.8	(−10.1)	33.9	(+2.4)

Table 7: Performance of automatically selected phrases (with corpus phrase filtering)

	Precision (% change) – 50 queries						
Recall	NL	tagged		tagged plus dictionary		with term filtering	
10	67.6	68.4	(+1.2)	69.4	(+2.7)	71.5	(+5.7)
20	54.3	57.3	(+5.6)	57.0	(+5.1)	59.1	(+8.8)
30	48.7	49.4	(+1.5)	49.5	(+1.7)	50.3	(+3.4)
40	42.5	41.4	(−2.5)	40.7	(−4.2)	42.1	(−0.8)
50	35.8	35.4	(−1.0)	36.0	(+0.8)	36.5	(+2.0)
60	28.3	29.0	(+2.3)	28.8	(+1.6)	29.3	(+3.6)
70	19.7	21.3	(+8.3)	21.6	(+9.5)	21.9	(+11.4)
80	15.7	17.0	(+8.4)	17.5	(+11.8)	17.8	(+13.4)
90	10.6	11.3	(+6.8)	11.7	(+10.4)	12.1	(+13.8)
100	8.0	8.5	(+6.0)	9.0	(+12.8)	9.3	(+16.4)
average	33.1	33.9	(+2.4)	34.1	(+3.1)	35.0	(+5.7)

Table 8: Performance of tagged phrases

	Precision (% change) – 50 queries				
Recall	NL	manual (unfiltered)		tagged+dictionary (filtered)	
10	67.6	68.5	(+1.3)	71.5	(+5.7)
20	54.3	58.6	(+8.0)	59.1	(+8.8)
30	48.7	51.0	(+4.6)	50.3	(+3.4)
40	42.5	43.7	(+2.7)	42.1	(−0.8)
50	35.8	38.4	(+7.4)	36.5	(+2.0)
60	28.3	31.9	(+12.6)	29.3	(+3.6)
70	19.7	23.7	(+20.4)	21.9	(+11.4)
80	15.7	18.0	(+14.6)	17.8	(+13.4)
90	10.6	12.8	(+20.2)	12.1	(+13.8)
100	8.0	10.2	(+27.3)	9.3	(+16.4)
average	33.1	35.7	(+7.7)	35.0	(+5.7)

Table 9: Performance of manual versus automatic phrase selection

tionship between those concepts in the text, but it is not clear what type of relationship. An important part of our work in the near future will be the design and implementation of interfaces appropriate for structured queries.

Acknowledgments

This research was supported in part by NSF Grant IRI-8814790 and by contract AFOSR 90-0110 with the Air Force Office of Scientific Research. The authors wish to thank Bob Krovetz and Penelope Sibun for discussions about this work. We also wish to thank Longman Group, Ltd. for making the LDOCE available, as well as Ken Church and Bell Laboratories for making available the stochastic tagger and bracketer.

References

Anick, P.G.; Brennan, J.D.; Flynn, R.A.; Hanssen, D.R.; Alvey, B.; Robbins, J.M. "A Direct Manipulation Interface for Boolean Information Retrieval via Natural Language Query", *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, 135-150, 1990.

Boguraev, B.; Briscoe, T. "Large Lexicons for Natural Language Processing: Utilizing the grammar coding system of LDOCE", *Computational Linguistics*, 13: 203-218; 1987.

Church, K. "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text", *Proceedings of the Second Conference on Applied Natural Language Processing*, 136-143, 1988.

Cleverdon, C.W.; Keen, E.M. *Factors Determining the Performance of Indexing Systems, Vol.1,2.* Cranfield, England. Aslib Cranfield Research Project, 1966.

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

Croft, W.B.; Das, R. "Experiments with query acquisition and use in document retrieval systems", *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, 349-368, 1990.

Das-Gupta, P. "Boolean Interpretation of Conjunctions for Document Retrieval", *Journal of the American Society for Information Science*, 38: 245-254; 1987.

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.

Fagan, J. *Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and Non-Syntactic Methods.* Ph.D. Thesis, Technical Report 87-868, Cornell University, Computer Science Department, 1987.

Gay, L.; Croft, W.B. "Interpreting Nominal Compounds for Information Retrieval", *Information Processing and Management*, 26(1), 21-38, 1990.

Fuhr, N. "Models for retrieval with probabilistic indexing", *Information Processing and Management*, 25, 55-72, 1989.

Fuhr, N.; Buckley, C. "Probabilistic Document Indexing from Relevance Feedback Data", *Proceedings of 13th ACM Conference on Research and Development in Information Retrieval*, 45-62, 1990.

Harman, D. "Towards interactive query expansion", *Proceedings of 11th ACM Conference on Research and Development in Information Retrieval*, 321-332, 1988.

Harman, D.; Candela, G. "Retrieving records from a gigabyte of text on a minicomputer using statistical ranking", *Journal of the American Society for Information Science*, 41, 581-589, 1990.

Krovetz, R.; Croft, W.B. "Lexical Ambiguity and Information Retrieval, *ACM Transactions on Information Systems*, (to appear).

Lewis, D. D. *Representation and Learning in Information Retrieval.* Ph.D. Thesis, in preparation, 1991.

Lewis, D.D.; Croft, W.B. "Term Clustering of Syntactic Phrases", *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, 385-404, 1990.

Pearl, J., *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, California, 1989.

Van Rijsbergen, C.J. *Information Retrieval.* Butterworths, London; 1979.

Robertson, S.E. "The Probability Ranking Principle in IR", *Journal of Documentation*, 33, 294-304, 1977.

Salton, G. *Automatic Information Organization and Retrieval.* McGraw-Hill, New York; 1968.

Salton, G.; McGill, M. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York; 1983.

Salton, G.; Fox, E.A.; Wu, H. "Extended Boolean Information Retrieval". *Communications of the ACM*, 26, 1022-1036, 1983.

Salton, G.; Yang, C.S.; Yu, C.T. "A Theory of Term Importance in Automatic Text Analysis", *Journal of the American Society for Information Science*, 26: 33-44; 1975.

Smeaton, A.; Van Rijsbergen, C.J. "Experiments on Incorporating Syntactic Processing of User Queries into a Document Retrieval Strategy". *Proceedings of ACM SIGIR International Conference on Research and Development in Information Retrieval*, 31-52, 1988.

Smith, M.E. *Aspects of the P-Norm Model of Information Retrieval: Syntactic Query Generation, Efficiency and Theoretical Properties.* Ph.D. Thesis, Computer Science Department, Cornell University, 1990.

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

Tong, R.M.; Shapiro, D.G. "Experimental Investigations of Uncertainty in a Rule-Based System for Information Retrieval", *International Journal of Man-Machine Studies*, 22: 265-282; 1985.

Turtle, H. *Inference Networks for Document Retrieval*, Ph.D. Thesis, University of Massachusetts, COINS Technical Report 90-92, 1990.

Turtle, H.R.; Croft, W.B. "Evaluation of an Inference Network-Based Retrieval Model", *ACM Transactions on Information Systems*, (to appear).