Experiments with Query Acquisition and Use in Document Retrieval Systems

W. Bruce Croft and Raj Das
Department of Computer and Information Science
University of Massachusetts, Amherst, MA. 01003

Abstract

In some recent experimental document retrieval systems, emphasis has been placed on the acquisition of a detailed model of the information need through interaction with the user. It has been argued that these "enhanced" queries, in combination with relevance feedback, will improve retrieval performance. In this paper, we describe a study with the aim of evaluating how easily enhanced queries can be acquired from users and how effectively this additional knowledge can be used in retrieval. The results indicate that significant effectiveness benefits can be obtained through the acquisition of domain concepts related to query concepts, together with their level of importance to the information need.

1 Introduction

One of the most successful techniques developed for information retrieval is *relevance feedback* (Salton, 1968). This technique was developed in the context of statistical document retrieval, but it has also appeared in a slightly different form for database retrieval (Tou et al, 1982). In its simplest form, relevance feedback is used after an initial set of documents have been retrieved by comparing them to the query with a statistical ranking function (Salton, 1968, Van Rijsbergen, 1979). The person who generated the query then examines the top ranked documents and identifies which of these documents are relevant and which are not. The words and word frequencies in these documents are then used to modify the initial query and a new document ranking is formed. Retrieval experiments have shown that significant improvements in effectiveness can be obtained in this manner (for example, Sparck Jones and Webster, 1980; Salton and Buckley, 1988). One way of explaining the effectiveness of this technique is that it provides a simple method for acquiring more of the user's knowledge and using it to refine the query by adding related words and changing the relative importance of words.

Relevance feedback does, however, have some drawbacks. The principal ones are:

1. Identifying documents as relevant is a very crude way of identifying the information in the documents that is of particular interest. Many words from relevant documents that are unrelated to the information need can be included in the query.

2. Relevance feedback does not improve the initial search. If either no relevant documents or very few are found in the initial ranked list, users will be less likely to be satisfied with the system's performance.

In the I^3R document retrieval system (Croft and Thompson, 1987), emphasis is placed on the acquisition of a detailed model of the information need (or query) through interaction with the user. Analysis of an initial query and domain knowledge provide the basis for this interaction. The domain knowledge is used both to assist in the analysis of the query and to find related words and concepts. Since domain knowledge is typically not available for many applications, facilities are provided for users to provide this knowledge during query formulation. In addition, the relevance feedback process in I^3R is enhanced in that users not only specify the relevance of retrieved documents, but also the particular words and concepts that are important and their relationships to other concepts in the domain knowledge base.

There is obviously a contrast between what is expected from the user of a traditional statistical relevance feedback system and a system such as I^3R. The dialogue between the I^3R system and the user is designed to elicit as much as possible of the user's knowledge of the concepts mentioned in the query and of other domain concepts related to them. In a traditional system, the interaction with the users is minimized in that the only information they provide is the initial query (preferably in natural language) and relevance judgments. Increasing the amount and the complexity of the interaction with the users carries a penalty in terms of the effort required. The expectation is that this penalty will be more than offset by substantial improvements in retrieval effectiveness relative to simpler systems.

As we move to experimental systems that use knowledge-based and natural language processing techniques for document retrieval, the ability to interactively acquire domain and linguistic knowledge from users becomes crucial. This is because, in any real application, the knowledge that these techniques require will be incomplete or missing entirely. Given that some researchers believe that users are not even capable of providing more than the simplest information, the determination of what type of knowledge can be acquired interactively and how it can be acquired is a fundamental issue.

Our general approach is to acquire knowledge iteratively through interaction with the user. We also believe that acquisition should be done in the context of particular queries. The goal of finding relevant documents should motivate users to provide knowledge to the system. This can be contrasted to the approach of having a separate knowledge acquisition phase before the system can be used for any queries as, for example, in the IRACQ or TELI systems (Ayuso et al, 1988, Ballard and Stumberger, 1986). The knowledge that is acquired, together with the initial query, can be regarded as an "enhanced" query. Our research has focused on the following specific types of knowledge:

- Relative importance of query concepts.
- Complex query concepts (phrases).
- Domain knowledge (concepts related to the initial query concepts).

Each of these types of knowledge will be discussed in the next section.

The experimental methodology described in section 3 is designed to test the hypothesis that users of an information system can provide knowledge during query formulation

that will improve retrieval effectiveness. The effectiveness of the enhanced queries will be compared both with simple queries and with queries modified using relevance feedback techniques. One of the major issues in designing a methodology is that the overall retrieval effectiveness can be affected by both the success of the techniques for acquiring enhanced queries from users, and the effectiveness of the retrieval techniques that use the additional knowledge in the enhanced query. In most cases, previous research and information retrieval models can be used to show why specific types of knowledge *should* improve the effectiveness of the system. A failure to obtain effectiveness improvements could then be regarded as a failure in knowledge acquisition. As some of the retrieval strategies we are using in this study are new, however, some changes to these strategies may be expected during the initial set of experiments and the separation of acquisition and use is not so straightforward.

Another feature of the methodology described here is that it is designed for evaluating complex, interactive retrieval systems. The evaluation of these types of systems using standard test collections has been recognized as a difficult problem, and as new, large collections of text become available for research, it becomes increasingly important to develop appropriate methodologies.

The fourth section of the paper contains the results of the experiments of experiments and a discussion. The appendix to the paper contains an example of the questionnaire that was used for query formulation.

2 Using Query Knowledge

2.1 Relative Importance of Query Concepts

The simplest type of knowledge a user can provide is to indicate, in a particular query, the words and phrases that are particularly important. In the I^{3}R system, this is done by using a pointing device to highlight these words and phrases in a natural language query (see Figure 1). For statistical document retrieval systems, this provides two types of information:

1. The relative importance of query words (or query term weights).
2. The important groups of words (phrases).

The way this information can be used is best described using the probabilistic model of retrieval, although the same information has been used effectively in systems based on the vector space model (Salton and McGill, 1983; Salton, 1986; Fagan, 1987). This section discusses the use of relative importance information, the next discusses phrases.

The probabilistic model of retrieval shows that the optimal ranking function for documents, given certain assumptions, is

$$P(Relevant|D)/P(NonRelevant|D)$$

where $P(Relevant|D)$ is the probability that a document is relevant given its representative D. Using Bayes' Rule we transform this ranking function to $R(D)/Q(D)$, where R(D) is the

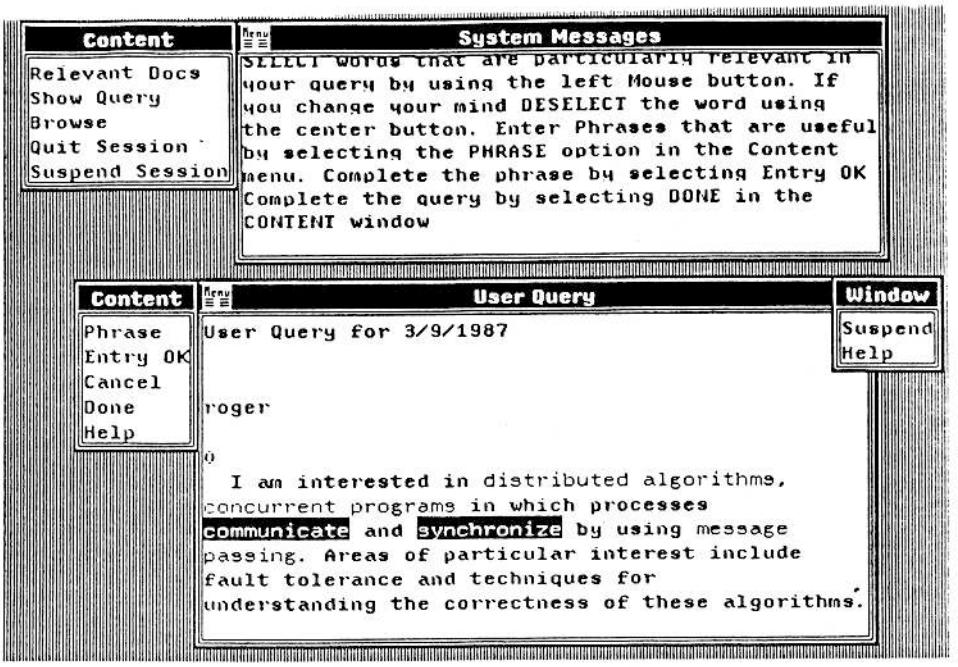

Figure 1: Selecting important words and phrases in I^{3}R.

probability that a relevant document has representative D, and $Q(D)$ is the probability of a non-relevant document having representative D. $R(D)$ and $Q(D)$ are usually expanded by assuming that the terms are independent in the relevant and non-relevant sets of documents (Van Rijsbergen, 1979). The *approximation* to $R(D)$ derived in this way is

$$R'(D) = \prod_{i=1}^{n} p_i^{t_i}(1-p_i)^{1-t_i},$$

where p_i is the probability that t_i is 1 in a random document from the relevant set of documents. A similar expression holds for $Q'(D)$, with the probability q_i being the probability that t_i is 1 in the non-relevant set of documents. Given these approximations, the ranking function can be shown to be

$$\sum_{i|qt_i=1} w_i\, t_i \tag{1}$$

where

$$w_i = \log\left[\frac{p_i(1-q_i)}{(1-p_i)q_i}\right]$$

and the summation is over all query terms The estimation of p_i and q_i can be done using information acquired during relevance feedback. Initially, however, this information is not

available and the estimation of these parameters from document statistics results in a ranking function where w_i is approximated by the *inverse document frequency weight*, measured by $\log nd/frequency(t_i)$ (Croft and Harper, 1979). The parameter nd is the number of documents stored and $frequency(t_i)$ is the number of documents that contain t_i (sometimes called the term posting). An approximation for nd that is often used is the maximum term posting.

Another form of ranking function (1) can be developed that includes the within-document frequency information and has better performance (Croft, 1981, 1983). This ranking function includes a probability called the *term significance weight* that can estimated by normalizing the within document frequency for a term in a particular document. This weight, intuitively, measures the importance of a term in a given document whereas the inverse document frequency weight measures the importance of the term in the whole collection of documents. The actual definition of the term significance weight is $P(t_i = 1|D)$, which is the probability that term i is assigned to document representative D. For term i in document j, the term significance weight is referred to by s_{ij} and the resulting ranking function is

$$\sum_{i|qt_i=1} s_{ij}\, w_i\, t_i \tag{2}$$

Ranking function (2) is equivalent to the "tf.idf" form of the ranking function derived from the vector space model. We are also carrying out experiments with other forms of retrieval models (Fuhr, 1989), but these studies are beyond the scope of this paper.

Information about the relative importance of query terms can be used to get better estimates for w_i prior to relevance feedback. Specifically, instead of making assumptions about p_i values that result in w_i being essentially equivalent to the inverse document frequency weight, the relative importance information can be used to provide better estimates for p_i. There has been some research that indicates that user-defined query weights can be used effectively, although these experiments had many limitations (Salton and Waldstein, 1978; Harper, 1980).

From the acquisition point of view, it is not clear how many levels of importance can be specified by the users. It does seem unlikely that they can reliably specify numeric probability values. In I^3R, there are only two levels: `important` and `default`. In Harper's thesis (1980), he describes an experiment where 5 levels of importance are simulated. In the experiments reported here, 4 levels are used.

2.2 Phrases

Improvements to retrieval models that make an assumption of term independence have been suggested by Van Rijsbergen (1979) and Yu (1983). In general, these approximations result in ranking function (2) being replaced by

$$\sum s_{ij}\, w_i\, t_i + A \tag{3}$$

where A is a correction factor applied to documents that contain dependent terms. Experiments with these models have not, in general, led to significant performance increases. In

another application of these models, Croft (1986) proposed using phrases identified in the query as dependent groups of terms. Documents that contain phrases receive an adjustment to the score they have obtained from retrieval strategy based on the independence model. The experiments using this approach, and the work by Smeaton on syntactic phrases, which used the same underlying retrieval model, showed improvements in retrieval performance (Smeaton and Van Rijsbergen, 1988). The details of the retreival model are given in the following subsection.

Another way of using phrases is to add terms representing phrases to the document and query vectors (Fagan, 1987). In Fagan's study, these extended vectors are then compared using a variation of the cosine correlation that separates the contributions of the words and phrases. Weights for the phrase terms are estimated from the weights of the words that make up the phrase rather than directly from the document collection. Fagan's results showed significant performance increases for some collections (such as CACM) and only small increases for others. Despite differences in implementation, the underlying use of phrases to correct a score resulting from an independence model is similar to the probabilistic approach. In our experiments, we have used both approaches.

2.2.1 A Probabilistic Phrase Model

If we restrict the phrases to two words, which has been shown to be the most effective in other studies, then we can use Van Rijsbergen's dependency model (1979) to calculate the correction factor in equation 3 as follows:

$$A = \sum (t_j \left[\log \left[\frac{(1-c_i)}{(1-d_i)}\right] - \log \left[\frac{(1-c_i')}{(1-d_i')}\right]\right] + t_i t_j \left[\log \left[\frac{c_i(1-d_i)}{(1-c_i)d_i}\right] - \log \left[\frac{c_i'(1-d_i')}{(1-c_i')d_i'}\right]\right])$$

where $t_i t_j$ is a dependent pair of terms, c_i, c_i' are $P(t_i = 1|t_j = 1)$ in the relevant and non-relevant sets of documents respectively, and d_i, d_i' are $P(t_i = 1|t_j = 0)$ in the relevant and non-relevant sets. The summation is over all pairs of terms identified as phrases. This could be rewritten as

$$A = \sum (t_j a_1 + t_i t_j a_2)$$

This makes it clear that documents containing both terms of a phrase receive a correction of $a_1 + a_2$. Documents containing only the second term in a phrase receive a correction of a_1. This can result in negative correction factors as we will see later.

We are then left with the problem of estimating c_i, c_i', d_i, d_i'. Similar to the way p_i and q_i are estimated, we can use the entire collection of documents to estimate the c_i', d_i' (the values in the non-relevant set), and estimate the values in the relevant set using constants in the initial retrieval and then a sample of relevant documents after relevance feedback. The maximum likelihood estimates are used in this study. They are

c_i' = (no. of co-occurrences of t_i, t_j)/(freq. of occurrence of t_j)
d_i' = (freq. of t_i - no. of co-occurrences)/ (size of collection - freq. of t_j)

	Phrase		
	t_1, t_2	t_1, t_2	t_3, t_4
Frequencies	191, 281	191, 281	140,226
No. of co-occurrences	31	31	13
c_i, d_i	.9, .6	.9, .3	.9, .4
c_i', d_i'	.11, .05	.11, .05	.06, .04
a_1	-0.57	-0.80	-0.77
a_2	0.37	0.91	0.92
A	-0.2	0.11	0.15

Table 1: Correction Calculation for 2 Phrases from CACM Collection (3730 documents)

An example of correction factors calculated for two phrases in the CACM collection (Salton, Fox and Wu, 1983) is shown in Table 1. Documents that contain only the second term of a phrase would receive a negative correction in each case. The first two columns show that the c_i, d_i estimates for relevant documents are important and can lead to negative corrections even for documents that contain both terms of a phrase.

2.3 Domain Knowledge

Knowledge-based systems rely on having a detailed model of the application domain. This domain knowledge can be quite complex, including arbitrary predicates on domain objects, causal relationships and temporal relationships. In the document retrieval application, however, our aim is to acquire this knowledge from the end users during retrieval sessions. This means that acquisition must be limited to types of knowledge that are both easily understood and directly applicable to retrieval. In the I^{3}R system and in this study, we aim primarily to capture the type of knowledge that is found in a thesaurus. More specifically, the domain knowledge is defined as concepts and relationships between them. The relationships we concentrate on are **is-a** (computer is-a device), **instance-of** (vax instance-of computer), **part-of** (processor part-of computer), **synonym-of** (program synonym-of software), and **related-to** (computer related-to hardware). The **related-to** relationship is very general and is used to describe relationships that could, in more detailed representations, be very complex. There are two approaches to acquiring this type of domain knowledge:

1. Ask the user to specify concepts that are related to query concepts and the types of relationships. This approach is very open-ended and does not provide much guidance. In I^{3}R, expert users are given this option both during query formulation and relevance feedback (Figure 2).

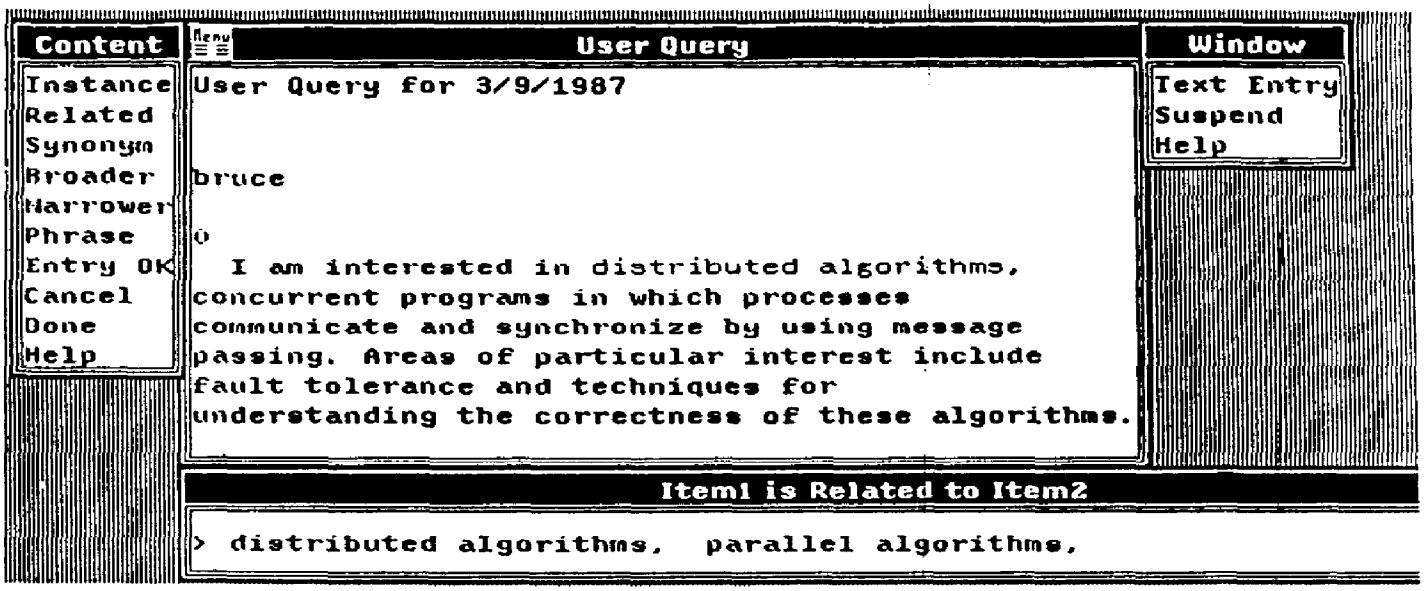

Figure 2: Domain knowledge specification in I^3R: user relates two concepts formed as phrases.

2. Suggest possible related concepts and ask the user to clarify relationship types and validate the relationship. This approach is easier for the user, but more difficult for the system. Possible sources of related concepts include term clusters and information from relevance feedback (Harman, 1988).

The domain knowledge that is acquired can be used in the retrieval process in a number of ways. The typical use of this knowledge in a statistical retrieval system would be to expand the query with related concepts. This has been done in previous projects with mixed results (Salton, 1968; Sparck Jones, 1971), and this is the technique used in our initial experiments.

Another way of using this knowledge is in retrieval based on plausible inference (Van Rijsbergen, 1986, Croft et al, 1989). In this approach, retrieval is viewed as establishing plausible relationships between the query and the documents, and assessing the degree of plausibility. Domain knowledge obviously helps establish these relationships and, as such, takes part in the matching process. Rather than simply using it to expand the query, however, it is used as a source of evidence that is combined with evidence from statistical and NLP sources. The difference between using domain knowledge for query expansion and

for plausible inference is clearer in the case where a significant domain knowledge base has been acquired. The knowledge base of concepts and relationships can be thought of as a network of nodes (representing concepts) and links (representing relationships). If we use this knowledge base to expand the query, then potentially every concept represented could be included if we follow all paths of links. The plausible inference process attempts to find connections between the concepts in the query and the concepts in particular documents. This approach is not pursued further in this paper.

2.4 Relevance Feedback

Relevance feedback provides much more information for the estimation of the probabilities in ranking function (2). The p_i estimates, in particular, can be improved greatly using the sample of relevant documents. It is also possible to modify the initial query by including words from the relevant documents. Salton and Buckley (1988) have recently done a series of experiments in which they establish that this type of query expansion, even by adding all words in the relevant documents, is very effective. In I^3R, it is also possible for the user to indicate important words and phrases in the text of relevant documents. Both techniques will be used in the experiments reported in this paper.

In order to establish that enhanced queries are more effective than simple queries, it is important to include relevance feedback. It may be, for example, that the words included through relevance feedback provide sufficient domain knowledge for effective retrieval. Our hypothesis is that enhanced queries will be more effective on the initial search and approximately the same effectiveness after feedback. This should result in more relevant documents being found overall.

3 Experimental Methodology

The approach used to acquire enhanced queries in the set of experiments reported here was to have people fill out a form designed for this purpose. The form was developed by looking at the performance and comments of a small group of users. Although there are disadvantages to this approach compared to acquiring queries through an interactive session with a retrieval system (for example, I^3R), we felt that it gave us more control and eliminated many factors that may affect performance, such as computer experience. We should emphasise that this form was designed to be used only in this study, and is not proposed as a subtitute for interactive query formulation. The lessons we learn from these experiments will be applied to the design of the next, larger study that will include interactive query formulation.

A limitation of our approach is the assumption that all users in the study have similar types of information needs. That is, we are assuming that all users want to see as many relevant documents as possible in the sample of documents they are shown, and that document abstracts can satisfy their need. This is a consequence of the artificial nature of the information needs in typical experimental settings.

A copy of one of the query forms we obtained in the study appears in the appendix. A person using this form is asked to provide some personal data and then increasingly detailed specifications of their information need. The initial query is a natural language statement of interest. The person then underlines important words and phrases in that query and indicates the importance of those concepts using a simple numerical level (1 to 3). This actually gives 4 levels of relative importance since those words not underlined are given a default value. The most complicated part of the form is a table where the person enters concepts from the initial query and then writes down related concepts, the relationship type, and the importance of the concepts. Finally, space is provided for criticism of the form and for comments on the difficulty of the specification process.

Once the form was filled out, the information from it was entered into a system which did the indexing and ran a variety of retrieval strategies based on the models discussed in the previous section. The top 10 documents in the ranking for each strategy were merged into a set in random order (to remove any bias toward the first documents seen) and shown to users for relevance judgments. Users were also asked to identify interesting concepts in the relevant documents. The relevance judgments were then used in a variety of feedback strategies, the top 10 documents for each strategy were merged (excluding documents already found by individual strategies), and this set was shown to users for more relevance judgments. Turnaround times were on the order of 2-3 days.

Comparison of the retrieval results is done using a matched-pair design (Robertson, 1981). In this design, the top ten documents in the ranking produced for a particular query by each pair of retrieval strategies are compared. The comparison, which simply identifies if one group of ten documents is better than the other, is on the basis of precision, or the number of relevant documents retrieved. This type of evaluation has the following advantages:

- It does not require full relevance judgments for each query. This is an important requirement for real system evaluation.
- It is realistic in the sense that users of a retrieval system will tend to examine only the top group of retrieved documents and are unlikely to make major distinctions based on the actual rankings in that top group. Traditional recall/precision tables are very sensitive to the initial rank positions and evaluate entire rankings.
- Significance measures can be readily used.

The disadvantage is, of course, that we do not obtain full recall/precision figures. Note that we can, however, estimate recall using the total number of relevant documents retrieved by all queries as has been done in many previous studies.

A total of 20 queries were processed for the experiments reported here. All experiments used a collection of abstracts from the *Communications of the ACM* between 1958 and 1985. Note that we are not using queries from the CACM test collection described in Salton, Fox and Wu (1983), and we have increased the coverage of the collection to include abstracts from 1979-1985.

Average number of terms in initial query (after stopword removal)	14.8
Average number of phrases in initial query	3.8
Average number of words in domain knowledge	9.4
Average number of phrases in domain knowledge	4.2

Table 2: Size of Queries and Domain Knowledge

Importance	% Use
Very Important	50.7%
Moderately Important	40.4%
Weakly Important	8.9%

Table 3: Use of Importance Levels for Words/Phrases

4 The Experiments

4.1 Acquisition Statistics

In this section, we present a number of statistics derived from the 20 forms that were filled out in the initial study. Of the 20 subjects, 5 were graduate students from a department of industrial engineering, 7 were graduate students from a computer science department, 5 were computer science undergraduates, 2 were graduate students from an electrical and computer engineeering department, and 1 was a graduate student from a department of education. Eleven of the subjects identified their degree of experience with the topic of the query as `intermediate`, 5 as `expert`, and 4 as `novice`. Eleven of the subjects have had some experience with IR systems, the other 9 have seldom or never used them.

Table 2 lists the average number of words and phrases that were provided as part of the initial query and the related domain knowledge. These figures clearly indicate that users (of the type in our study) are capable of providing enhanced queries. Table 3 lists the average use of the relative importance values used for words and phrases. The fact that the usage is heavily skewed towards the top two levels suggests that people may only distinguish a small number of levels of importance.

Table 4 shows the use of the relation types in the domain knowledge. This data shows a fairly even split in usage between **is-a** (in the form of broader-term and narrower-term), **synonym**, and **instance-of**. The other forms of relations were only used infrequently.

Finally, 11 of the subjects mentioned that they had some difficulty with query formulation. Most of the problems centered on the parts of the query form associated with the specification of domain knowledge. Based on their responses, the most difficult part of

Relation Types	% Use
Synonym	35.5%
Broader-Term	15.5%
Narrower-Term	13.7%
Instance-Of	18.4%
Part-Of	8.2%
Related-To	8.7%

Table 4: Use of Relation Types in Domain Knowledge

query formulation (for 5 of the 20 subjects) was identifying relation types in the domain knowledge.

4.2 The Initial Search

In this set of experiments, six retrieval strategies based on the models described in section 2 were used. These were:

Strategy 1: The baseline search using the independence model with tf.idf weights, as described in section 2.1.

Strategy 2: A search using phrases as described in section 2.2. The "phrases" were obtained using Fagan's approach, which is to use all pairs of words in the query and weight phrases by an average of the tf.idf scores of the individual terms.

Strategy 3: Same as strategy 1, but with p_i estimates based on user-specified relative importance values (0.5 for default, 0.6 for weakly important, 0.75 for moderately important, 0.9 for very important).

Strategy 4: A phrase search using the dependence model described in section 2.2.1, user-specified phrases, and estimates of p_i, c_i, d_i based on user-specified importance values.

Strategy 5: Same as strategy 4, but with words and phrases from domain knowledge included. Weights for domain knowledge words and phrases based on user-specified importance values.

Strategy 6: Same as strategy 3, but with words from domain knowledge (no phrases). Weights based on user-specified importance values.

The results are shown in Table 5. The average number of relevant documents found for each query by all strategies was 7.1. The average precision of the top 10 documents for each strategy is shown in Table 5. These figures show a 30% increase in precision for strategy 6 (including domain words) compared to the tf.idf baseline. We then carried out a series of

Strategy	Precision (20 queries)
Strategy 1	.37
Strategy 2	.35
Strategy 3	.40
Strategy 4	.35
Strategy 5	.41
Strategy 6	.48

Table 5: Results for Initial Searches

significance tests using a one-tailed sign test with $\alpha = .05$ (Siegel, 1956). The major results are as follows:

1. Neither of the phrase strategies were significantly different to the tf.idf strategy, or to each other. This suggests that we do not really understand how to use phrases. This is emphasised by the relative performance for strategies 5 and 6, where using words alone produced significantly better performance.

2. The use of user-specified importance weights with words (strategy 3) is significantly better than tf.idf at a level of .055.

3. The inclusion of domain knowledge leads to very significant performance increases. For example, in comparing strategy 6 with tf.idf, 12 of the queries performed better with strategy 6, and the other 8 queries had the same performance. Our experiments indicated that the user importance weighting was an important part of the success of this strategy.

4.3 Relevance Feedback

The main aim of these experiments was to see if a tf.idf search in combination with feedback significantly outperformed an enhanced query strategy with feedback. Five different relevance feedback strategies were used:

Strategy 7: Based on tf.idf search (strategy 1), all terms in relevant documents are added to initial query, and p_i estimates are based on occurrences in relevant documents. This strategy was designed to be similar to strategies described in Salton and Buckley (1988) that included all relevant document terms.

Strategy 8: Based on strategy 3, adding all terms from relevant documents to the initial query.

Strategy 9: Based on strategy 6, adding all terms from relevant documents to the initial query.

Strategy	Precision (20 queries)
Strategy 7	.20
Strategy 8	.22
Strategy 9	.23
Strategy 10	.29
Strategy 11	.38

Table 6: Results for Relevance Feedback

Strategy 10: Based on strategy 5, adding only words and phrases indentified by users in relevant documents to the initial query, and estimating p_i, c_i, d_i from relevant documents.

Strategy 11: Based on strategy 6, adding only words identified by users in relevant documents to the initial query.

The average number of relevant documents found for each query by all feedback strategies was 6. The average number of words identified by users in all relevant documents seen was 30.3. The average number of phrases identified was 12.9. The average precision for each feedback strategy is shown in Table 6. The major results from these experiments are:

1. Adding all terms from relevant documents was not as effective as using only words identified by the users. This is not the result obtained by Salton and Buckley (1988) and indicates that further work needs to be done to better understand the process of automatic query expansion.

2. Feedback strategy 11 was very effective. This means that not only does the enhanced initial query perform significantly better than tf.idf, it continues to perform better after feedback.

Another interesting piece of data is obtained by looking at the overlap between the words in the domain knowledge provided during query formulation and the words in the relevant documents used for feedback in strategy 7. Only 35% of the extra words provided by users were found in the relevant documents. This indicates that users are a potentially valuable source of domain knowledge and that a small sample of relevant documents will not necessarily contain the words that are important for describing the information need. It also indicates, however, that relevant documents are a good source of words to suggest to the users (Harman, 1988).

5 Conclusions

The results of this study indicate that enhanced queries significantly improve the effectiveness of retrieval strategies. The users in our study were able to provide a large amount

of knowledge about the topic of their information needs, and retrieval strategies were able to make effective use of this knowledge. The most useful types of knowledge that were obtained (in terms of retrieval effectiveness) were domain concepts related to the query concepts, and the relative importance of concepts. The use of phrases was not successful. Other experiments indicate that this is caused by the inability of the phrase-based search strategies to make effective use of this knowledge.

Our future work will involve a larger study using interactive query formulation, rather than forms. We also intend to continue to study phrase-based search strategies, including the evaluation of new retrieval models for phrases with standard test collections.

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.

References

Ayuso, D.; Shaked, V.; Weischedel, R. "An Environment for Acquiring Semantic Information." *Proceedings of the Twenty-Fifth Annual Meeting of the Association for Computational Linguistics*, 32-40; 1987.

Ballard, B.; Stumberger, D. "Semantic Acquisition in TELI: A transportable User-Customized Natural Language Processor". *Proceedings of the 24th Meeting of the Association for Computational Linguistics*, 20-29, 1986.

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

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.; Harper, D.J. "Using probabilistic models of document retrieval without relevance information", *Journal of Documentation*, 35, 285-295, 1979.

Croft, W.B.; Thompson, R.T. "I^{3}R: A New Approach to the Design of Document Retrieval Systems". *Journal of the American Society for Information Science*, 38: 389-404; 1987.

Croft, W.B., Lucia, T.; Cringean, J,; Willett, P. "Retrieving Documents by Plausible Inference: An Experimental Study". *Information Processing and Management*, 25, 599-614, 1989.

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

Computer Science Department, 1987.

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

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

Harper, D.J. *Relevance Feedback in Document Retrieval Systems: An Evaluation of Probabilistic Strategies.* Ph.D. Thesis, Computer Laboratory, University of Cambridge, 1980.

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

Van Rijsbergen, C.J. "A Non-Classical Logic for Information Retrieval". *Computer Journal*, 29: 481-485; 1986.

Robertson, S.E. "The methodology of information retrieval experiment". In: Sparck Jones, editor, *Information Retrieval Experiment.* London: Butterworths, 9-31, 1981.

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.; Buckley, C. "Improving Retrieval Performance by Relevance Feedback". Technical Report, Cornell University, 1988.

Salton, G; Waldstein, R.G. "Term Relevance Weights in On-Line Information Retrieval", *Information Processing and Management*, 14, 29-35, 1978.

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

Siegel, S. *Nonparametric Statistics*, McGraw-Hill, 1956.

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.

Sparck Jones, K. *Automatic Keyword Classification for Information Retrieval.* Butterworths, London, 1971.

Sparck Jones, K.; Bates, R.G. *Report on a design for the 'ideal' test collection.* British

Library Report 5428, Computer Laboratory, University of Cambridge, 1977.

Sparck Jones, K.; Webster, C. *Research on Relevance Weighting.* British Library Report 5553, Computer Laboratory, University of Cambridge, 1980.

Tou, F.M. et al. "RABBIT: An intelligent database assistant". *Proceedings AAAI-82*, 314-318, 1982.

Yu, C.T.; Buckley, D.; Lam, K; Salton, G. "A generalized term dependence model in information retrieval". *Information Technology: Research and Development*, 4, 129-154, 1983.

Appendix: A Sample Query Form

Query Formulation Questionnaire

1. About the Questionnaire

This questionnaire asks you to create a query to a database of articles from the computer science journal *Communications of the ACM* (CACM). The database contains titles, abstracts, and related information on articles that appeared in CACM between 1958 and 1985. We would like you to think of a topic in computer science that you would be interested in reading CACM articles on, and write a query describing that topic. By a query we mean a description of the content of articles you are interested in rather than, for instance, the names of authors. The following is an example of a query:

> I would like papers about information retrieval that address issues in distributed databases. Also of interest to me would be articles about the use of parallel architectures in implementing retrieval (particularly commercial) systems.

We will be asking you to give a number of additional pieces of information about your query, as well as some data about yourself. This information will be used to retrieve titles and abstracts of CACM articles that are meant to address your interest. After this information has been retrieved, we will present you with a list of these titles and abstracts and ask you to judge whether each of them is in fact relevant to your query.

2. Personal Data

Name: *Jane Doe*
Age : *29*
Sex : *Female*
Dept: *Computer & Information Science*
Your degree of experience with the topic of your query
(E)XPERT, (I)NTERMEDIATE, (N)OVICE: *E, I*
Your degree of experience with using text retrieval systems (i.e. computerized library catalogs, online bibliographic databases, etc.)
(N)EVER, (SE)LDOM, (SO)METIMES, (F)REQUENT: *SO*

3. The Actual Query

In the space provided below please print or type your query:

I am interested in articles about knowledge [3] based natural [3] language processing. I am also interested in articles that discuss the use of connectionist [2] techniques within the knowledge-based natural language [3] processing paradigm.

4. Important Words and Phrases in the Query

Please underline the important words and phrases of the query.

5. Providing a Ranking

Please categorize the words/phrases that you just underlined into one of three levels of importance. Put the appropriate number on top of the underlined words/phrases. If you feel that the word/phrase you underlined is very important to your query, please write down 3. If it is moderately important write down 2, and if less important write down 1.

6. Providing Related Words and Phrases

Please provide words/phrases that are related to the ones you underlined. For each **underlined** word/phrase in the query, please write down the word/phrase under the first heading in the following diagram, and list the related words/phrases under the second heading.

Filling out the Relation Type and Importance Level Headings in the diagram will be described in Sections 7 & 8 respectively

Word/Phrase from Query	Related Words/Phrases	Relation Type	Importance Level
Knowledge-based	*semantic*	*Narrower-Term*	1
Knowledge-based	*pragmatic*	*Narrower-Term*	1
Knowledge-based	*meaning-based*	*Synonym*	2
Natural Language Processing	*Natural Language Understanding*	*Synonym*	3
Natural Language Processing	*NLU*	*Synonym*	3
Natural Language Processing	*NLP*	*Synonym*	3
Natural Language Processing	*Text Understanding*	*Part-Of*	3
Natural Language Processing	*Parsing*	*Instance-Of*	3
Natural Language Processing	*Conceptual Analysis*	*Instance-Of*	3
Connectionist	*Neural Network*	*Synonym*	3
Connectionist	*Artificial Neural Network*	*Synonym*	3
Connectionist	*ANN*	*Synonym*	3
Connectionist	*Relaxation Network*	*Instance-Of*	3
Connectionist	*Backpropagation Network*	*Instance-Of*	3

7. Identifying Relation Types

Please identify the relation types that exist between each of the words/phrases under the first heading of the table above and the related words/phrases you wrote under the second heading. Please choose one of the following relation types, and write it down under the third heading above. A few examples of the different relation types listed below are given below.

The Relation Types are:

SYNONYM, BROADER-TERM, NARROWER-TERM, INSTANCE-OF, PART-OF

If none of these relations appear to be appropriate, please use the generic relation RELATED-TO.

Examples of the different relation types:

Two-Dimensional Array would be a SYNONYM for Matrix
Data Structure would be a BROADER-TERM for Linked Lists
Mouse would be a NARROWER-TERM for Pointing Devices
VAX780 would be an INSTANCE-OF of Computers
Keyboard would be PART-OF a computer

8. Providing a Ranking

Under the importance level heading in section 6 please categorize the extra words/phrase you provided into one of three levels of importance. If you feel that the extra word/phrase you provided is very important to

your query, please write down 3. If it is moderately important write down 2, and if less important write down 1. This is identical to the information provided in Section 5, except that the importance levels are now being sought for the extra words/phrases.

9. Comments About the Questionnaire

Was this questionnaire difficult to fill out? Please feel free to comment on anything else about this questionnaire. For instance, was there any particular question that was difficult, confusing, ambiguous, etc.?

It would have been better if the table in section 6 were designed so that there was more space to fill in the related words for each word/phrase from the query.