Combining Automatic and Manual Index Representations in Probabilistic Retrieval

T.B. Rajashekar
National Centre for Science Information
Indian Institute of Science
Bangalore 560 012, India

W. Bruce Croft
Computer Science Department
University of Massachusetts
Amherst, MA 01003

Abstract

Results from research in information retrieval have suggested that significant improvements in retrieval effectiveness can be obtained by combining results from multiple index representations, query formulations, and search strategies. The inference net model of retrieval, which was designed from this point of view, treats information retrieval as an evidential reasoning process where multiple sources of evidence about document and query content are combined to estimate relevance probabilities. In this paper, we use a system based on this model to study the retrieval effectiveness benefits of combining the types of document and query information that are found in typical commercial databases and information services. The results indicate that substantial real benefits are possible.

1 Introduction

With the enormous growth in the number and size of bibliographic, full text and other electronic information sources, information service providers are under constant pressure to provide their users with the most relevant items of information, in response to their information needs. Each information item in these databases has several clues (properties or content representations) about relevance in the form of natural language text (e.g., title, abstract, full text), manually assigned index terms, subject categories, etc. Similarly, a variety of clues can be obtained from the users about their information needs (e.g., natural language descriptions, term importance, known relevant papers, etc.). In Appendix 1, the relevant portions of a completed user profile information sheet used in a SDI service are shown, to illustrate the way this information is obtained (Rajashekar, 1988).

The descriptions of information needs are typically used to construct Boolean queries for Boolean logic (or exact-match) retrieval systems. While these systems are quite effective for some kinds of searching (e.g., known-item searching), when it comes to more general searching or for untrained users, they often result in either no output, not enough output, or too much output (Cooper, 1988; Maron, 1988). To address these problems, systems based on "best-match" retrieval models have been developed which rank the retrieved documents by a score which is based on the probability of relevance of the document to the query. The best known of such models are the vector space and probabilistic retrieval models (Salton & McGill, 1983; Bookstein, 1985; Belkin & Croft, 1987; Turtle & Croft, 1990).

Systems based on best-match retrieval typically support simple natural language queries and automatic document indexing. This type of system has consistently performed much better than the exact-match techniques under laboratory conditions using test collections of a few thousand records, and we have begun to see the commercialization of these techniques and evaluation of their effectiveness with large text databases (Wagers, 1992; Callan & Croft, 1993; Harman & Candela, 1991; Callan et al., 1992).

Many of the experiments that have been done with best-match systems have used very simple representations of documents and queries relative to what is available in an operational setting, as described above. Some results have been obtained, however, using multiple representations. These results show that a) a given query will retrieve different documents when applied to different representations, even when the average retrieval performance (recall/precision) achieved with each representation is similar (Katzer et al., 1982; Croft &

Harper, 1979), b) documents retrieved by multiple representations are more likely to be relevant (Katzer et al., 1982; Fox et al., 1988; Croft et al., 1989), c) given a single natural language description of an information need, different searchers will formulate different queries to represent different aspects of that need and will retrieve different documents, even when average performance is similar for each searcher (Katzer et al., 1982; McGill et al., 1979; Saracevic & Kantor, 1988), and d) documents retrieved by multiple searchers and search strategies are more likely to be relevant (Saracevic & Kantor, 1988; Turtle & Croft, 1991a; Belkin et al., 1993).

These results indicate that significant improvements in retrieval effectiveness may be possible if we can combine results obtained by using multiple document representations and query strategies. By adopting retrieval techniques that support this capability, operational retrieval systems can better exploit the variety of document and query clues that already exist.

Recently, an inference network-based probabilistic retrieval model has been proposed which views information retrieval as an evidential reasoning process in which multiple sources of evidence about document and query content are combined to estimate the probability that a given document matches an information need (Turtle & Croft, 1990). 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. INQUERY, a retrieval system based on this model, supports sophisticated indexing and complex query formulation (Callan et al., 1992). INQUERY has been used successfully on a variety of text databases ranging up to a few gigabytes in size.

In the study reported in this paper, our research goal was to demonstrate the flexibility of the inference net model in combining manual and automatic index representations in documents and user queries. The study is an extension of other experiments with INQUERY that have used multiple query and document representations (Turtle & Croft, 1991a; Belkin et al., 1993; Callan & Croft, 1993). The significance of the experiments presented here, compared to the previous work, is that they use representations that are commonly available in information services, they investigate more combinations of representations, and they examine the simultaneous use of multiple query and document representations. This work also extends the original work of Katzer (Katzer et al., 1982) in that more combinations of representations are studied and, more importantly, does this in the context of a retrieval model designed for combining representations.

In more specific terms, we report the results of a series of experiments conducted using INQUERY to evaluate the following hypotheses:

1. Significant improvements in retrieval effectiveness can be obtained by combining multiple document representations for a given representation of the information need.
2. Significant improvements in retrieval effectiveness can be obtained by combining results from multiple index representations in queries.
3. Significant improvements in retrieval effectiveness can be obtained by combining results from multiple query types.

Our interest here is in index representations for subject access like controlled vocabulary terms, classification codes, subject headings, indexer selected terms and phrases from natural language text (keywords), automatically generated index terms, etc. The "query type" mentioned in the third hypothesis refers to the way the query is expressed in the INQUERY language. Examples of query types are Boolean queries, simple natural language queries, queries containing phrases, and queries with weighted terms.

In section 2 we briefly describe the probabilistic inference net model which is the basis of our experiments. In section 3 we describe the experimental methodology. In section 4 we present the experimental results and a discussion of these results. Section 5 contains the conclusions.

2 Probabilistic Inference Network Retrieval Model

The experiments described in sections 3 and 4 were carried out using the INQUERY retrieval engine developed at the Information Retrieval Laboratory in the University of Massachusetts. In what follows, we give a brief description of the inference net model on which this system is based, and the INQUERY query language. More details of INQUERY can be found in (Callan et al., 1992) and the inference net model in (Turtle, 1990; Turtle & Croft, 1991b; Turtle & Croft, 1991a). In this paper, the emphasis will be on the ability of the model to handle multiple sources of evidence.

The inference net model is a probabilistic retrieval model in that it follows the Probability Ranking Principle. 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 (Robertson, 1977). 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. The inference net model is based on Bayesian inference networks (Pearl, 1988). These are directed, acyclic dependency graphs (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. In other words, the matrix specifies $P(q\ is\ true|p\ is\ true)$, $P(q\ is\ true|p\ is\ false)$, $P(q\ is\ false|p\ is\ true)$, and $P(q\ is\ false|p\ is\ false)$. 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 network, these networks can be used to compute the probability or degree of belief associated with all remaining nodes.

Fig. 1 shows the basic document retrieval inference network used in INQUERY. It consists of two component networks : one for documents and one for queries. The document network is built once for a collection and the structure does not change during query processing. It consists of document nodes (d_j's) and concept representation nodes (r_m's). The concept representation nodes or representation nodes can be divided into several subsets, each corresponding to a single representation technique that has been applied to the document texts. For example, if the phrase "information retrieval" has been extracted automatically and "information retrieval" has been manually assigned as an index term, then two representation nodes with distinct meanings will be created. We represent the assignment of a specific representation concept to a document by a directed arc to the representation node.

Each representation node contains a specification of the conditional probability associated with the node given its set of parent nodes. While, in principle, computation of this probability would require $O(2^n)$ space for a node with n parents, since we only consider one document at a time in this model, a simple estimation formula can be used. The probability estimate that is used (Turtle & Croft, 1991a) is very similar to the *tf.idf* weights used in

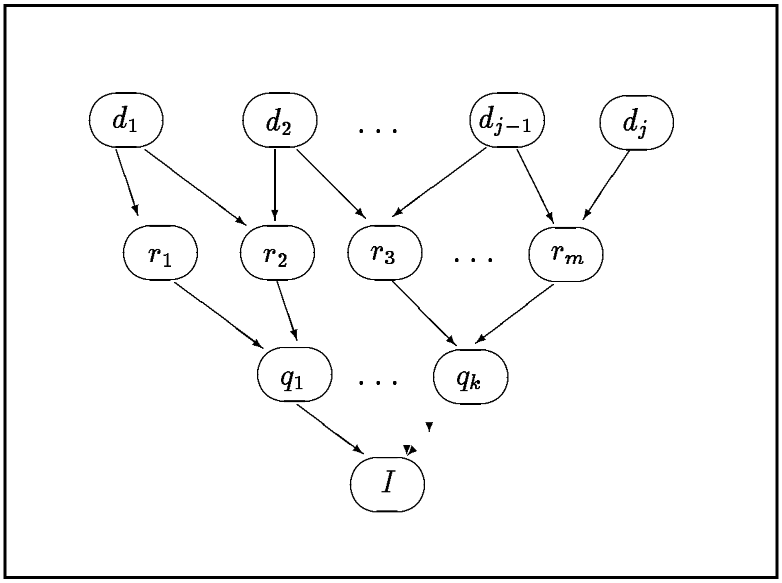

Figure 1: Basic document inference network

many previous IR experiments (Salton & McGill, 1983). The actual formula used was

$$0.4 + 0.6 * (0.4 + 0.6 * \frac{\log(tf + 0.5)}{\log(maxtf + 1.0)}) * \frac{\log(\frac{collectionsize}{f})}{\log(collectionsize)} \qquad (1)$$

where tf is the frequency (of the term associated with the representation node) in the document, $maxtf$ is the maximum term frequency in the document, and f is the number of documents in which the term occurs.

The query network is an "inverted" DAG with a single leaf node (I) representing the user's information need, one or more representations of that information need (q_k's), and multiple roots that correspond to the concept representation nodes. The q_k nodes are used in cases where multiple queries are used to represent the information need. Examples of this are when both a Boolean query and a simple natural language query are used (Turtle & Croft, 1991a), or when multiple versions of a query are generated by search intermediaries (Belkin et al., 1993).

A query is processed by constructing the query network and attaching it to the document network. The attachment of the query concept nodes to the document network has no effect on the structure of the document network. The probability that the information need is met given a particular document d_j is computed by setting the value of the d_j node to *true* and computing the probabilities associated with each node in the query network,

OPERATOR	ACTION
#and	Specifies that all terms should be present in the document.
#or	Specifies that any of the specified terms should be present.
#not	Specifies that the term should not be present.
#sum	Specifies that the more terms present, the better.
#wsum	Specifies that some terms are more important than others, but also that the more present, the better.
#max	Specifies that the term with the highest associated probability is the most important. A form of synonym operator.
#n	Specifies that the terms should be present, in order, with at most $n-1$ words between them.
#uw$n$	Specifies the terms should be present, in any order, in a text window of size n.
#phrase	Specifies that the terms should be present as a simple noun phrase. The way that probabilities are combined for these terms depends on the statistics for the phrase in the document collection (Croft et al., 1991).
#syn	A synonym operator where all terms are treated as equivalent.

Table 1: The operators in the INQUERY query language.

given this evidence. Probabilities are computed using the indexing weight specifications (term frequency, inverse document frequency, etc.) associated with the representation nodes. This is repeated for each document in the network and the probabilities used to rank the documents. Simplifying assumptions made for efficient implementation of inference networks, their construction and evaluation, are discussed in (Turtle & Croft, 1991b).

The INQUERY query language provides a set of operators to specify concepts and relationships between concepts. This language is not designed to be used by searchers directly but rather is a target language for the user interface. In terms of the inference net model, the query language operators specify the structure and node types of the query network. All the operators, therefore, combine probabilities from parent nodes. In an inference net model, the concepts represented by the parent nodes are treated as independent sources of evidence for the new concept defined by the query operator. The #and operator, for example, is a probabilistic version of Boolean AND that combines the parent node probabilities by multiplying them. This follows from the assumption that the concept represented by the #and only represents a document when all parent concepts also represent that document. Table 1 lists the current set of operators. New operators can be added to represent different linguistic structures or relationships.

These operators use simplified expressions to calculate the probability for the associated node in the query network. The derivation of these expressions and their use are given in (Turtle, 1990). For a query node Q with parents P_1,...,P_n where P(P_1 = *true*)=p_1,...,P(P_n = *true*)=p_n, the expressions for some of the operators are :

$$
\begin{aligned}
\mathrm{P}_{\mathrm{not}}(Q) &= 1 - p_1 && (2)\\
\mathrm{P}_{\mathrm{or}}(Q) &= 1 - (1-p_1)\cdot\ldots\cdot(1-p_n) && (3)\\
\mathrm{P}_{\mathrm{and}}(Q) &= p_1 \cdot p_2 \cdot \ldots \cdot p_n && (4)\\
\mathrm{P}_{\mathrm{max}}(Q) &= \max(p_1, p_2, \ldots, p_n) && (5)\\
\mathrm{P}_{\mathrm{wsum}}(Q) &= \frac{(w_1p_1 + w_2p_2 + \ldots + w_np_n)w_q}{(w_1 + w_2 + \ldots + w_n)} && (6)\\
\mathrm{P}_{\mathrm{sum}}(Q) &= \frac{(p_1 + p_2 + \ldots + p_n)}{n} && (7)
\end{aligned}
$$

The #n, #uw$n$, #phrase and #syn operators calculate probabilities based on the statistics of words in the documents. In a sense, they create new indexing concepts. In the case of #n and #uw$n$, the number of occurrences of words satisfying the proximity restriction are used to calculate a *tf.idf* probability, as with simple concept representation nodes. For example, the number of occurrences of the term #3(information retrieval) in the individual documents and the collection are used as the *tf* and *f* values in the *tf.idf* probability formula (1). The #syn operator uses the total occurrences for all the terms mentioned as the tf and f values.

The #phrase operator is more complicated in that it is treated differently based on the statistics of the word co-occurrences (Croft et al., 1991). The first case, where the phrase is common, results in the #phrase operator being the same as the #3 operator. The second case, where the phrase is moderately frequent, results in the #phrase operator being treated as #max(#3(terms) #sum(terms)). The third case, where the phrase is rare, results in the #phrase operator being ignored and the terms are simply included into the operator that included the #phrase. For example, #sum(t1 #phrase(t2 t3)) would become #sum(t1 t2 t3). The statistics that determine which of the three cases apply include the mutual information measure (MIM) for the terms in the phrase (van Rijsbergen, 1979) and the frequency of the #3 occurrences. Specifically, the first case happens when MIM $>$ 3.0 and frequency > 1250. The second case happens when MIM > 1.25 and frequency > 30.

Henceforth in this paper, we will use 'index representation' to refer to both document and query representations. The context of usage will clarify whether we are referring to 'index representations' in queries or documents. Furthermore, we will use 'query type' and 'query representation' synonymously.

The operators given in Table 1 provide a powerful means for combining different index representations and query types. For example, a Boolean query and a simple natural language query could be combined using the #sum operator as,

#sum(*Boolean Query* #sum(*Natural language*))

Similarly, different index representations of a query, for example using controlled vocabulary terms, keywords, and simple natural language, can be combined as,

#sum(#sum(*Thesaurus terms*) #sum(*Keywords*) #sum(*Natural language*))

Furthermore, the #wsum operator can be used to assign weights to individual query or index representations within these combined representations.

3 Experimental Methodology

The experiments reported in this paper were carried out using standard IR methodology in which a test collection consisting of documents, queries, and relevance judgements for each query, is used to generate recall-precision figures (Sparck Jones & van Rijsbergen, 1976). Comparisons of retrieval effectiveness are made using tables of precision values at ten standard recall points (i.e., 10% of relevant documents retrieved, 20% ...100%)), averaged over a set of queries, for each of the query/index representations and their combination being evaluated. When two tests are being compared, we show the difference as the percentage change from the baseline test. A difference of 5 percent on average is generally considered significant, and a 10 percent difference is considered very significant (Sparck Jones & Bates, 1977). These "rules-of-thumb" are based on differences that would be noticeable for a user. Standard tests of significance have generally not been used in recall-precision evaluations because of doubts about their validity (van Rijsbergen, 1979). In these experiments, we used a sign test based on the differences in average precision for each of the 50 queries. With few exceptions, the significance test supported the rules-of-thumb in that a difference of 5%

or more in the average precision as shown in the recall-precision tables was significant at the .05 level. The exceptions to this are noted in the discussion of the experiments.

The experimental results are generally presented as full recall-precision tables. In the case of the comparison of document representations, however, where there are 12 experiments involved, only the average precision over the 10 recall points is reported. This is done for conciseness, and the general trends are so clear that the relative performance levels at high or low recall points are less important.

A primary requirement for this study was the availability of a test collection supporting multiple representation types. We selected the INSPEC test collection (Katzer et al., 1982) for this study as it supports multiple document representations - controlled vocabulary terms, keywords (indexer selected significant terms and phrases from document titles and abstracts) and the natural language text of titles and abstracts themselves. The INSPEC subject categories would have been an interesting additional index representation to use for this study, but unfortunately the test collection records did not include this representation. The INSPEC test collection contains 12,684 records covering the areas of computer, electrical and electronic engineering, 84 queries in natural language and standard relevance judgements. Out of the 84 queries in the test collection, we selected 50 queries for this study. This selection was made based on the clarity of their expression, enabling accurate identification of key concepts and construction of various query strategies required for the study. Selecting the INSPEC test collection for this study has the additional advantage that it is representative of the bibliographic databases used in many of the operational information retrieval service centres, and the observations and conclusions reached in this study would therefore be that much more appropriate for such settings. Most other test collections available for retrieval experiments do not contain manual indexing (the controlled vocabulary terms and keywords).

We first generated the following basic automatic and manual index representations and query types required for this study :

A. Index Representations :

1. Queries :

   - Automatic index representation : Indexing each stem in the query text (*Tx*). This is done by removing stopwords and stemming the remaining words (Salton & McGill, 1983). Note that the indexing is carried out at search time.

- Manual index representation :
  - (a) Analysis and representation of query concepts using thesaurus terms (*Th*), and
  - (b) Analysis and representation of query concepts using keywords, i.e., terms and phrases manually identified from the natural language query (*KW*). During this analysis, the keywords were also assigned weights which were used to formulate weighted term queries (see below).

2. Documents :
   - Automatic index representation : Indexing each stem in the title and abstract text fields (*Tx*). This is the same process as automatic query indexing.
   - Manual index representation : Indexing each word in thesaurus terms (*Th*) and keywords (*KW*).

B. Query Types :

1. Natural language query formulated as a probabilistic query using the #sum operator (*Tx*).
2. Boolean query, formulated using keywords and the Boolean operators #and, #or and #not (*BOOL*).
3. Weighted term query, formulated as a probabilistic weighted sum query using keywords and the #wsum operator (*WTERM*). Two sets of weighted term queries were generated using a scale of two and three importance levels - most important (1.0) and less important (0.5), and most important (1.0), moderately important (0.5) and less important (0.3).

It may be noted that within the index representations in queries, multi-word terms in keywords and thesaurus terms were represented as phrases using the #phrase operator. In the subsequent sections of the paper we will use the abbreviations shown inside the brackets as short hand notation to refer to their respective representations.

Many operational information retrieval systems either support some or all of these index and query representations or possess enough details from which these representations can be easily generated (see, for example, Appendix 1). These individual representations are combined to generate specific combinations required for evaluating the research hypotheses of this study. Details of specific combinations produced are discussed in Section 4. We conducted three sets of experiments corresponding to the three hypotheses. In the first set of experiments, we compared the performance of single index representations in queries (Th, KW, Tx) on the document file indexed on one, two and three sources of evidence (Th, KW, Tx). In the second set of experiments, while keeping the sources of evidence in the document file the same (a combination of Th, Tx and KW), we compared the performance of combined index representations in queries generated by a combination of two (Th,Tx; Th,KW; Tx,KW) and three index representations (Th,Tx,KW). In the third set of experiments we compared the performance of individual query types (Tx, BOOL, WTERM) with their combined representation (Tx,BOOL; Tx,WTERM). The results of these experiments are presented in the following section.

As an example of the types of queries that were produced, the following is the original text of one of the queries in the test collection.

> I am interested in the area of document representation in information retrieval, particularly controlled vocabulary systems. Anything on controlled vocabularies (i.e. thesauri, subject index terms) would be useful but other things on document representation might be as well for comparative purposes.

The following variations of this query were produced manually using the original text and, for the Th queries, the thesaurus as the source vocabulary.

1. The Tx version of this query simply puts a #sum operator around this text. Stemming and stopword removal takes place when the query is processed.

2. The query formulated using controlled vocabulary terms (Th) was as follows:

   ```
   #sum( indexing,
   #phrase(information retrieval systems),
   #phrase(information retrieval),
   thesauri,
   vocabulary )
   ```

3. The query formulated using keywords (KW) was:

   ```
   #sum( #phrase(document representation),
   #phrase(information retrieval),
   ```

```
#phrase(controlled vocabulary systems),
thesauri,
#phrase(subject index terms) )
```

4. The Boolean version of the query (BOOL) was:

```
#and(
#phrase(document representation)
#phrase(information retrieval)
#or(
#phrase(controlled vocabulary)
thesauri
#phrase(subject index terms) ) )
```

5. One of the weighted sum versions (WTERM) was:

```
#wsum( 1.0
1.0 #phrase(document representation),
0.5 vocabulary,
0.5 thesauri,
0.5 #phrase(subject index terms),
0.5 #phrase(information retrieval) )
```

6. Combinations of these representations were done using the #sum or #wsum operator. For example, the following is the query for the combined Boolean and natural language queries:

```
#wsum( 1.0
1.0 #and(#phrase(document representation) #phrase(information retrieval)
#or( #phrase(controlled vocabulary) thesauri #phrase(subject index terms) ) )
1.0 #sum( I am interested in the area of document representation in information
retrieval, particularly controlled vocabulary systems. Anything on controlled vocab-
ularies - thesauri, subject index terms would be useful but other things on document
representation might be as well for comparative purposes. ) )
```

	Document Index Files (Collection Size : 12684 docs.)						
	Th	KW	Tx	Th,KW	Th,Tx	KW,Tx	Th,KW,Tx
Unique Stems	1851	9722	17983	9840	18068	18323	18383
Max stem frequency	3162 (comput)	4821 (system)	11833 (system)	7508 (system)	14520 (system)	16654 (system)	19341 (system)
Stem occurrences	61872	144146	579290	206018	641162	723436	785308
Postings	56975	125889	417592	155119	444834	426836	450342
Max within doc freq	8	10	32	10	32	32	32

Table 2: Summary of collection statistics

4 Experimental Results

We discuss the results in terms of the three hypotheses.

Hypothesis 1 : Significant improvements in retrieval effectiveness can be obtained by combining multiple document representations for a given representation of the information need.

To test this hypothesis, we used three query files, each file consisting of the 50 queries represented in a specific index representation. Queries in the first two files were formulated using the manual index representations 'Th' and 'KW' and the third file consisted of the natural language queries (Tx) providing the automatic index representation. Probabilistic sum (operator #sum) was used as the search strategy.

Seven document inference network files were generated using the INSPEC records :

1. Three files of single source of evidence - Thesaurus (Th), keywords - indexer selected terms and phrases from title and abstracts (KW) and natural language text (titles and abstracts) (Tx),

2. Three files of two sources of evidence - Th,KW; Th,Tx; KW,Tx, and

3. A combined file of all the three sources of evidence - Th,KW,Tx.

The collection statistics for these seven index files is shown in Table 2.

Each of the three query files was processed on the corresponding single evidence (e.g., 'Th' query file on 'Th' index file), two evidence (e.g., 'Th' query file on 'Th,Tx' and 'Th,KW' index files) and the combined three evidence (e.g., 'Th' query file on 'Th,Tx,KW' index file) index files, and the results compared with the standard relevance judgements. A summary

QUERIES	SOURCES OF EVIDENCE (Documents)						
	Single Evidence			Two Evidences			Combined
	Th	KW	Tx	Th,KW	Th,Tx	KW,Tx	Th,KW,Tx
Th	8.8	-	-	12.1 (+37.8)	14.2 (+61.4)	-	15.1 (+71.1)
KW	-	16.5	-	18.7 (+13.7)	-	26.1 (+58.2)	27.9 (+69.1)
Tx	-	-	22.3	-	24.0 (+7.9)	24.3 (+9.3)	25.3 (+13.7)

Table 3: Single and multiple sources of evidence in documents

of the results obtained is given in Table 3. The figures shown are average precision obtained over ten standard recall points. Figures inside the brackets are percentage improvements obtained by use of two and three sources of evidence in the document file, over the results obtained using a single source of evidence.

From Table 3 it can be seen that there is generally a significant improvement in retrieval effectiveness as we move from the use of single to multiple sources of evidence in the document file, while the number of sources of evidence in the query remains unaltered. The only improvement in this table that was not rated as significant by the sign test was the Th,Tx combination compared to Tx on its own.

The results clearly show that controlled vocabulary terms are not an effective representation on their own. They also show, however, that their presence as an additional source of evidence in the document file can contribute to the improved performance of queries formed using other index representations. This is evident if we compare the figures (Table 3) for 'KW' and 'Tx' queries on 'KW,Tx' and 'Th,KW,Tx' document index representations. While the evidence provided by thesaurus terms by themselves is quite weak, their presence in the documents improved the performance of KW and Tx queries.

Hypothesis 2 : Significant improvements in retrieval effectiveness can be obtained by combining results from multiple index representations in queries.

We evaluated the first hypothesis by using different combinations of automatic and manual index representations as sources of evidence in the document file and studied their retrieval performance on queries expressed in a single index representation. To test the second hypothesis, we combined manual and automatic index representations in queries and studied their retrieval performance on the same document file. We generated four query files, using the following combinations of index representations :

1. Thesaurus and keyword queries (Th,KW),

2. Thesaurus and natural language queries (Th,Tx),

3. Keyword and natural language queries (KW,Tx), and

4. Combined query file of Th,KW, and Tx (Th,KW,Tx).

We used the same query files (i.e., Th, KW and Tx) that were used in the first set of experiments to generate these combinations. Within each file, different index representations of a query were combined using the #sum operator. For example, the format of the combination of a query expressed as Tx and using the thesaurus terms would be #sum(#sum(Th) #sum(Tx)).

These four query files were processed on the combined document index file of 'Th,KW,Tx' and the results evaluated with the standard relevance judgements. We used the combined document index file for these tests as this had produced the best results in the first set of experiments. The results are given in Tables 4,5 and 6. In each of these tables, the figures in the second column are for the queries expressed in a single index representation, the figures in third and fourth columns are for the queries expressed as a combination of two index representations and the figures in the fifth column are for the combination of three index representations.

As general observations, it may be seen from the results shown in Tables 4, 5 and 6 that

1. Queries formulated by manually selecting words and phrases outperform the simple natural language queries (Column 2 of Table 5 and Table 6). This performance improvement comes from two sources; the removal of unnecessary words and the use of phrases (Croft et al., 1991).

2. Adding automatic index representations (i.e., natural language query Tx) to manual index representations (i.e., keywords and thesaurus terms) in queries significantly improves the performance of these representations (Column 4 of Table 4 and Table 5). Note that the result with keywords is significant at the .05 level even though the improvement is slightly less than 5%.

3. The performance of controlled vocabulary terms in queries can be significantly improved by combining them with either the natural language query or keywords selected from the query (Table 4).

4. The best overall performance was obtained by combining the natural language queries with manually selected keywords (Column 4 of Table 6 and Table 5). Adding controlled vocabulary terms to these queries reduced performance.

	Precision (% change) – 50 queries			
Recall	Th	Th,Tx	Th,KW	Th,KW,Tx
10	34.8	49.6 (+42.2)	52.2 (+49.7)	58.7 (+68.5)
20	28.1	42.8 (+52.5)	45.9 (+63.4)	50.1 (+78.6)
30	23.1	35.3 (+52.7)	40.4 (+74.9)	43.4 (+87.6)
40	18.5	29.0 (+56.7)	31.9 (+72.4)	35.3 (+90.3)
50	14.4	22.5 (+56.0)	25.9 (+79.7)	28.3 (+95.9)
60	11.3	18.9 (+67.0)	20.7 (+82.9)	23.8(+111.0)
70	9.0	13.8 (+52.5)	16.1 (+77.8)	18.2(+101.7)
80	7.1	10.6 (+48.1)	11.6 (+63.4)	13.4 (+88.4)
90	3.6	6.1 (+71.2)	6.5 (+81.9)	8.2(+130.3)
100	0.8	2.4(+186.3)	3.0(+254.5)	3.3(+287.0)
average	15.1	23.1 (+53.1)	25.4 (+68.5)	28.3 (+87.4)

Table 4: Combining thesaurus terms with keyword and automatic index representations in queries

	Precision (% change) – 50 queries			
Recall	KW	Th,KW	KW,Tx	Th,KW,Tx
10	64.3	52.2 (−18.8)	65.2 (+1.5)	58.7 (−8.6)
20	53.6	45.9 (−14.4)	54.7 (+2.1)	50.1 (−6.5)
30	41.7	40.4 (−3.1)	45.5 (+9.2)	43.4 (+4.0)
40	31.9	31.9 (+0.1)	34.1 (+6.8)	35.3 (+10.6)
50	27.1	25.9 (−4.3)	27.8 (+2.7)	28.3 (+4.3)
60	22.3	20.7 (−7.2)	23.3 (+4.5)	23.8 (+7.0)
70	16.9	16.1 (−5.2)	17.8 (+5.2)	18.2 (+7.6)
80	12.1	11.6 (−3.9)	13.2 (+8.8)	13.4 (+10.8)
90	7.1	6.5 (−9.4)	7.9 (+10.9)	8.2 (+14.7)
100	1.6	3.0 (+82.4)	2.0 (+24.7)	3.3 (+99.2)
average	27.9	25.4 (−8.8)	29.2 (+4.6)	28.3 (+1.4)

Table 5: Combining keywords with thesaurus and automatic index representations in queries

When the last point is examined in more detail, the results show that addition of thesaurus terms to automatic index representations (Tx) improved precision at middle and high recall points, while lowering precision at low recall points. Their addition to keywords lowered precision at all recall levels except the highest. But when all three index representations were combined in the queries, thesaurus terms helped in improving precision at middle and

	Precision (% change) – 50 queries			
Recall	Tx	Th,Tx	KW,Tx	Th,KW,Tx
10	63.8	49.6 (−22.3)	65.2 (+2.3)	58.7 (−7.9)
20	50.2	42.8 (−14.8)	54.7 (+8.9)	50.1 (−0.2)
30	38.2	35.3 (−7.5)	45.5 (+19.4)	43.4 (+13.7)
40	28.6	29.0 (+1.4)	34.1 (+19.0)	35.3 (+23.2)
50	23.5	22.5 (−4.4)	27.8 (+18.2)	28.3 (+20.0)
60	17.9	18.9 (+5.4)	23.3 (+30.0)	23.8 (+33.2)
70	13.3	13.8 (+3.8)	17.8 (+34.3)	18.2 (+37.3)
80	9.9	10.6 (+6.6)	13.2 (+33.1)	13.4 (+35.6)
90	5.9	6.1 (+2.6)	7.9 (+33.4)	8.2 (+38.0)
100	1.8	2.4 (+35.8)	2.0 (+14.9)	3.3 (+83.6)
average	25.3	23.1 (−8.8)	29.2 (+15.2)	28.3 (+11.7)

Table 6: Combining automatic index terms with thesaurus and keywords in queries

high recall levels, while lowering precision at the top two recall levels. To see why this was happening, we looked at the probabilities (belief estimates) produced by these three representations and noticed that the probabilities produced by thesaurus terms were much higher than that produced by keywords and Tx. These higher probabilities seem to be produced due to the low collection frequencies of these terms in the test collection resulting in high inverse document frequencies. Consequently, when the rankings are combined, documents retrieved by thesaurus terms, which include both relevant and non relevant documents, tend to dominate the relevant documents retrieved by other index representations. We reformulated the combined index representation query as a weighted sum query (#wsum operator) and ran a series of experiments lowering the weight of thesaurus queries. The best performance was achieved when the thesaurus term queries were scaled by a factor of 0.3. The result is significantly better than any of the other combined query representations. The results are given in Table 7. Similar results have been reported with respect to ACM CR classification categories in the CACM test collection (Turtle, 1990).

Hypothesis 3 : Significant improvements in retrieval effectiveness can be obtained by combining results from multiple query types.

The query types we investigate here are the natural language queries (Tx), Boolean queries (BOOL) and weighted term queries (WTERM). The rationale for using these types for evaluating this hypothesis is that most operational retrieval systems use Boolean queries and these are usually constructed from the natural language description of the user's information

	Precision (% change) – 50 queries	
Recall	Th,KW,Tx	Th,KW,Tx (Th 0.3)
10	58.7	66.5 (+13.3)
20	50.1	55.1 (+10.0)
30	43.4	47.8 (+10.3)
40	35.3	38.1 (+8.1)
50	28.3	32.2 (+14.0)
60	23.8	25.0 (+5.1)
70	18.2	19.8 (+8.5)
80	13.4	14.0 (+4.5)
90	8.2	8.7 (+6.4)
100	3.3	2.6 (−19.5)
average	28.3	31.0 (+9.7)

Table 7: Reducing the weight of thesaurus terms

needs. By way of additional information that can facilitate Boolean query formulation, many of these systems also collect from the user a list of terms to be used for searching, and the importance they attach to these terms. Given this, we felt it would be interesting to find out the improvements that can be obtained by combining these query types.

We constructed Boolean queries using the query texts and combined these as separate queries with Tx queries using the #sum operator. Boolean, Tx and their combined representations were then processed separately on the combined document index file. The results are given in Table 8.

	Precision (% change) – 50 queries		
Recall	BOOL	Tx	Combined
10	55.6	63.8 (+14.8)	59.7 (+7.5)
20	44.4	50.2 (+13.2)	47.9 (+7.9)
30	37.5	38.2 (+1.7)	40.6 (+8.1)
40	29.8	28.6 (−3.9)	32.3 (+8.4)
50	25.9	23.5 (−9.2)	27.2 (+5.0)
60	21.3	17.9 (−15.9)	22.6 (+6.3)
70	17.1	13.3 (−22.4)	18.1 (+6.1)
80	10.6	9.9 (−6.7)	13.9 (+30.6)
90	6.4	5.9 (−6.9)	8.4 (+31.2)
100	1.5	1.8 (+15.1)	1.9 (+23.4)
average	25.0	25.3 (+1.2)	27.3 (+9.0)

Table 8: Combining Boolean and Tx query types

The results show significant improvements from the combination. In earlier experiments with the inference net model, however, much better improvements have been reported for the CACM collection (Turtle & Croft, 1991a). The difference is probably mostly due to the nature of the test collections, in that the INSPEC collection is much larger. Another way of looking at these results is that, in the absence of Boolean queries, similar results can be obtained by probabilistic processing of Tx queries alone.

In Table 8, the interpretation of Boolean queries is probabilistic, which has been shown to perform much better than exact-match interpretation in earlier experiments (Turtle, 1990). In Table 9 we show the difference between the exact-match (E-BOOL) and probabilistic interpretation of Boolean queries (P-BOOL) used in these experiments, for the INSPEC test collection.

	Precision (% change) – 50 queries		
Recall	E-BOOL	P-BOOL	
10	39.1	55.6	(+42.3)
20	29.4	44.4	(+51.0)
30	23.7	37.5	(+58.3)
40	15.9	29.8	(+87.1)
50	11.1	25.9	(+133.1)
60	8.4	21.3	(+154.6)
70	5.5	17.1	(+209.9)
80	2.2	10.6	(+378.6)
90	0.9	6.4	(+605.9)
100	0.4	1.5	(+278.3)
average	13.7	25.0	(+83.1)

Table 9: Exact match and probabilistic interpretation of Boolean queries

We constructed a weighted sum query (#wsum) of keywords by assigning weights to individual keywords on a scale of two importance levels - very important and less important, based on a careful analysis of the natural language queries. This weighted sum query was combined with the Tx as a separate query using the probabilistic sum operator. The results of processing these three query files (WTERM, Tx and the combined query strategy file) on the combined document index file is given in Table 10. It can be seen that significant improvements can be obtained by combining these query types.

While constructing the weighted term queries, we also considered whether different scales of term weights made any difference to search results. In addition to assigning weights on a scale of two importance levels (very important and less important), we also separately

	Precision (% change) – 50 queries		
Recall	WTERM	Tx	Combined
10	64.3	63.8 (−0.8)	68.3 (+6.2)
20	52.1	50.2 (−3.6)	57.0 (+9.4)
30	44.0	38.2 (−13.4)	47.7 (+8.4)
40	34.1	28.6 (−16.2)	37.0 (+8.4)
50	28.7	23.5 (−18.1)	30.5 (+6.2)
60	23.0	17.9 (−22.1)	24.2 (+5.1)
70	17.5	13.3 (−24.2)	18.4 (+5.3)
80	13.5	9.9 (−26.5)	14.6 (+8.7)
90	7.5	5.9 (−21.0)	7.9 (+6.0)
100	1.7	1.8 (+6.2)	2.0 (+20.5)
average	28.7	25.3 (−11.7)	30.8 (+7.4)

Table 10: Combining weighted term and Tx query types

assigned three level weights - very important, moderately important and less important. The results are given in Table 11. It appears that a scale of two weights perform as well as a scale of three weights. These results also show that the inclusion of weights results in only very small (and not significant) improvements relative to the unweighted keyword query. This is consistent with a result reported in (Croft & Das, 1990) and suggests that more experiments are required to determine if user-supplied weights are an important part of query formulation.

	Precision (% change) – 50 queries		
Recall	KW(No weights)	KW (3 weights)	KW(2 weights)
10	64.3	64.2 (−0.1)	64.3 (+0.1)
20	53.6	51.8 (−3.3)	52.1 (−2.7)
30	41.7	42.8 (+2.7)	44.0 (+5.6)
40	31.9	34.2 (+7.1)	34.1 (+7.0)
50	27.1	28.4 (+5.0)	28.7 (+6.1)
60	22.3	22.7 (+1.8)	23.0 (+3.2)
70	16.9	17.1 (+1.3)	17.5 (+3.5)
80	12.1	13.4 (+10.6)	13.5 (+11.2)
90	7.1	7.4 (+4.4)	7.5 (+5.2)
100	1.6	1.7 (+2.4)	1.7 (+2.1)
average	27.9	28.4 (+1.9)	28.7 (+2.8)

Table 11: Two and three levels of term importance

5 Conclusion

Based on the results in Section 4, we can accept the first two hypotheses that, by treating manual and automatic index representations in queries and documents as sources of evidence, significant improvements in retrieval effectiveness can be obtained by combining these sources of evidence in the inference net probabilistic retrieval model. We can also accept the third hypothesis that by combining different query types we can obtain results that are much better compared to using them on their own. The best performance was obtained using all three representations (controlled vocabulary, keywords and text) for both queries and documents, with the relative contribution of the controlled vocabulary representation downweighted. User weighting of query terms was not shown to have a significant benefit.

These results are consistent with, and complementary to, early investigations of similar hypotheses (Katzer et al., 1982; Saracevic & Kantor, 1988; Turtle, 1990; Belkin et al., 1993). The fact that the results reported in this paper for the combined index and query representations are much better than those reported in earlier experiments using the INSPEC test collection (Salton & Buckley, 1988; Fox & Koll, 1988) supports the view that the INQUERY framework provides a very effective way of implementing this approach to retrieval.

We believe these results have practical implications for operational information retrieval systems in the sense that by adapting probabilistic retrieval techniques they could more fully exploit the different 'clues' that exist in documents and natural language descriptions of user information needs. The perceived computational complexity of best-match retrieval models have been an hindrance for their use in large scale information services until recently (Rajashekar, 1988). Given the processing capabilities of present day workstations, the availability of inexpensive storage options, and the demonstrated efficient implementation of these models (Turtle & Croft, 1991b; Harman & Candela, 1991), the situation is ripe for wider use of these techniques.

Acknowledgments

This work was supported in part by a UNDP fellowship held by the first author at the University of Massachusetts, Amherst, U.S.A. during Sept 1992 to Feb 1993. Additional support was provided by the NSF Center for Intelligent Information Retrieval at Amherst. The authors acknowledge the very useful suggestions made by the reviewers.

References

Belkin, N., Cool, C., Croft, W., & Callan, J. (1993). The effect of multiple query representations on information retrieval system performance. In *Proceedings of ACM SIGIR International Conference on Research and Development in Information Retrieval*, pages 339–346.

Belkin, N. J. & Croft, W. B. (1987). Retrieval techniques. In Williams, M. E. (Ed.), *Annual Review of Information Science and Technology*, pages 109–145. Elsevier Science Publishers.

Bookstein, A. (1985). Probability and fuzzy-set applications to information retrieval. In Williams, M. E. (Ed.), *Annual Review of Information Science and Technology*, pages 117–151. Knowledge Industries Publications, Inc.

Callan, J. P. & Croft, W. (1993). An evaluation of query processing strategies using the TIPSTER collection. In *Proceedings of ACM SIGIR International Conference on Research and Development in Information Retrieval*, pages 347–356.

Callan, J. P., Croft, W., & Harding, S. (1992). The INQUERY retrieval system. In *Proceedings of the 3rd International Conference on Database and Expert Systems Applications*, pages 78–83.

Cooper, W. S. (1988). Getting beyond Boole. *Information Processing and Management*, 24(3):243–248.

Croft, W. B. & Das, R. (1990). Experiments with query acquisition and use in document retrieval systems. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 349–368.

Croft, W. B. & Harper, D. J. (1979). Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, 35:285–295.

Croft, W. B., Lucia, T. J., Cringean, J., & Willett, P. (1989). Retrieving documents by plausible inference: An experimental study. *Information Processing and Management*, 25(6):599–614.

Croft, W. B., Turtle, H., & Lewis, D. (1991). The use of phrases and structured queries in information retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 32–45.

Fox, E. A. & Koll, M. B. (1988). Practical enhanced Boolean retrieval: Experiences with the SMART and SIRE systems. *Information Processing and Management*, 24(3):257–267.

Fox, E. A., Nunn, G. L., & Lee, W. C. (1988). Coefficients for combining concept classes in a collection. In *Proceedings of the Eleventh Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 291–308, New York, NY. ACM.

Harman, D. & Candela, S. (1991). Retrieving records from a gigabyte of text on a minicomputer using statistical ranking. *Journal of the American Society for Information Science*, 41(8):581–589.

Katzer, J., McGill, M. J., Tessier, J. A., Frakes, W., & DasGupta, P. (1982). A study of the overlap among document representations. *Information Technology: Research and Development*, 1:261–274.

Maron, M. E. (1988). Probabilistic design principles for conventional and full-text retrieval systems. *Information Processing and Management*, 24(3):249–255.

McGill, M., Koll, M., & Noreault, T. (1979). An evaluation of factors affecting document ranking by information retrieval systems. Technical report, Syracuse University, School of Information Studies.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers.

Rajashekar, T. B. (1988). Improving SDI systems : Implications of new retrieval models. *Library Science with a Slant to Documentation*, 25:44–62.

Robertson, S. E. (1977). The probability ranking principle in IR. *Journal of Documentation*, 33(4):294–304.

Salton, G. & Buckley, C. (1988). Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(3):513–524.

Salton, G. & McGill, M. J. (1983). *Introduction to Modern Information Retrieval.* McGraw-Hill.

Saracevic, T. & Kantor, P. (1988). A study of information seeking and retrieving. iii. searchers, searches, overlap. *JASIS*, 39(3):197–216.

Sparck Jones, K. & Bates, R. G. (1977). Research on automatic indexing 1974–1976. Technical report, Computer Laboratory, University of Cambridge.

Sparck Jones, K. & van Rijsbergen, C. J. (1976). Information retrieval test collections. *Journal of Documentation*, 32(1):59–75.

Turtle, H. & Croft, W. (1991a). Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3):187–222.

Turtle, H. & Croft, W. B. (1990). Inference networks for document retrieval. In Vidick, J.-L. (Ed.), *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, pages 1–24. ACM.

Turtle, H. & Croft, W. B. (1991b). Efficient probabilistic inference for text retrieval. In *Proceedings RIAO 3*, pages 644–661.

Turtle, H. R. (1990). *Inference Networks for Document Retrieval.* PhD thesis, University of Massachusetts at Amherst.

van Rijsbergen, C. (1979). *Information Retrieval.* Butterworths, second edition.

Wagers, R. (1992). DowQuest and Dow Jones Text-Search : Which works best and when? *Online*, 16(6):35–41.

Appendix 1

Portions of a completed SDI profile form are reproduced here illustrating the variety of 'clues' obtained from the users about their information needs.

NATIONAL CENTRE FOR SCIENCE INFORMATION
Indian Institute of Science, Bangalore - 560 012 (India)

S.D.I. User Profile Information Sheet

Profile No. : B0110

1. *Area of Investigation :* Viral vaccines; Tissue culture vaccines.

2. *Project Title :* Development of Japanese Encephalitis Virus Vaccine.

3. *Project Details (Pl. underline important terms) :*

(a) *Objectives :*

- Development of Killed Japanese Encephalitis virus vaccine using tissue culture source.
- Studies on the development of live attenuated JE virus vaccine.
- Comparison of immunological response of JE virus vaccine against Indian strains.

(b) *Methods adopted, instruments used, applications envisaged :*

- Use of formalin for virus inactivation.
- Concentration of vaccine by ultrafilteration.
- Immunisation of humans/animals.

(c) *Any other useful information regarding the project :*

- Chick embryo culture(CEC),Vero, MKTC and BHK-21 cell cultures used for vaccine purpose.
- Safety tests of vaccine in vivo/in vitro.
- Potency assay of vaccines.

4. *Give titles and references of any two published papers, which are directly relevant to your project :*

(a) Singh, B., I.K. Chin Chang and W. McD. Hammon (1973). Semi-commercial scale production of JBE virus vaccines from tissue culture [Applied Microbiol, 25(6), 945-51].

(b) Guskey, Louis E. and Howard M. Jenkin (1975). Adaptation of BHK-21 cells to growth in shaker culture and subsequent challenge by JBE virus [Applied Microbiol, 30(3), 433-38].

5. *List below Subject Terms to be used for computer search :*

Most important terms	Alternate terms/ Synonyms
1. Japanese encephalitis	Encephalitis, Japanese B JE JBE
2. Tissue culture	Cells, cultured Cell culture, Cell line Primary culture Suspension culture
3. Mosquito-borne virus/virion/viruses	Arbovirus Flavivirus
4. Vaccine/Vaccines	Vaccination Immunisation

Other terms	Alternate terms/ Synonyms
5. Mass cultivation	Mass culture Mass production
6. Inactivated	Inactivation Killed
7. Attenuated	Attenuation Live
8. Adjuvants	Immunoadjuvants Immunologic adjuvants Immunoactivators Immunopotentiators Potentiators
9. Stabilises	
10. Potency assay	Bioassay Biological assay Immunoassay
11. Epidemic	
12. Replication, Virus	
13. Concentration, Virus	
14. Purification, Virus	
15. Lyphilisation, Virus	
16. Haemagglutination	Hemagglutination
17. Titration, Virus	
18. Complement fixation	
19. Plaque reduction	
20. Neutralisation	

6. *Suggest which of the above terms are to be considered alone and which to be considered together for searching. Give as many associations as you like. Indicate the associations using term numbers. For e.g., 1 and 2; 2,3 and 6; 2,4 and 9; etc.*

1,2; 2,3; 2,4; 1,5; 1,6; 1,7; 1,10; 1,11; 1,12; 1,13; 1,14; 1,15;
1,16; 1,17; 1,18; 1,19; 1,20; 3,5; 3,6; 3,7; 4,8; 4,9;

7. *How many papers do you expect to be published in your research area per month?*

   Less than ten.