TREC-2 Routing and Ad-Hoc Retrieval Evaluation using the INQUERY System

Bruce Croft, James Callan, and John Broglio
Computer Science Department
University of Massachusetts
Amherst, MA. 01003

1 Project Goals

The ARPA TIPSTER project, which is the source of the data and funding for TREC, has involved four sites in the area of text retrieval and routing. The TIPSTER project in the Information Retrieval Laboratory of the Computer Science Department, University of Massachusetts, Amherst (which includes MCC as a subcontractor), has focused on the following goals:

- Improving the effectiveness of information retrieval techniques for large, full-text databases,
- Improving the effectiveness of routing techniques appropriate for long-term information needs, and
- Demonstrating the effectiveness of these retrieval and routing techniques for Japanese full text databases [4].

Our general approach to achieving these goals has been to use improved representations of text and information needs in the framework of a new model of retrieval. This model uses Bayesian networks to describe how text and queries should be used to identify relevant documents [6, 3, 7]. Retrieval (and routing) is viewed as a probabilistic inference process which compares text representations based on different forms of linguistic and statistical evidence to representations of information needs based on similar evidence from natural language queries and user interaction. Learning techniques are used to modify the initial queries both for short-term and long-term information needs (relevance feedback and routing, respectively).

This approach (generally known as the inference net model and implemented in the INQUERY system) emphasizes retrieval based on combination of evidence. Different text representations (such as words, phrases, paragraphs, or manually assigned keywords) and different versions of the query (such as natural language and Boolean) can be combined in a consistent probabilistic framework. This type of “data fusion” has been known to be effective in the information retrieval context for a number of years, and was one of the primary motivations for developing the inference net approach.

Another feature of the inference net approach is the ability to capture complex structure in the network representing the information need (i.e. the query). A practical consequence of this is that complex Boolean queries can be evaluated as easily as natural language queries and produce ranked output. It is also possible to represent "rule-based" or "concept-based" queries in the same probabilistic framework. This has led to us concentrating on automatic analysis of queries and techniques for enhancing queries rather than on in-depth analysis of the documents in the database. In general, it is more effective (as well as efficient) to analyze short query texts than millions of document texts. The results of the query analysis are represented in the INQUERY query language which contains a number of operators, such as #SUM, #AND, #OR, #NOT, #PHRASE, and #SYN. These operators implement different methods of combining evidence and describing concepts.

Some of the specific research issues we are addressing are morphological analysis in English and Japanese, word sense disambiguation in English, the use of phrases and other syntactic structure in English and Japanese, the use of special purpose recognizers (for example, company, country and people name recognizers) in representing documents and queries, analyzing natural language queries to build structured representations of information needs, learning techniques appropriate for routing and structured queries, techniques for acquiring domain knowledge by corpus analysis, and probability estimation techniques for indexing.

The first TREC evaluation and the two previous TIPSTER evaluations have made it clear that a lot remains to be learned about retrieval in large, full-text databases based on complex information needs. Issues as phrases, relevance feedback, and probability estimation have proven to be quite difficult in such environments. On the other hand, the effectiveness levels achieved have been quite good. The experiments done in the TREC-2 evaluation, together with the 24 month TIPSTER evaluation which followed it, were designed to improve our understanding about which IR techniques work and why.

2 System Description

The document retrieval and routing system that has been developed on the basis of the inference net model is called INQUERY [2]. The main processes in INQUERY are **document indexing**, **query processing**, **query evaluation** and **relevance feedback**.

In the document indexing process, documents are parsed and index terms representing the content of documents are identified. INQUERY supports a variety of indexing techniques including simple word-based indexing, indexing based on part-of-speech tagging and phrase identification, and indexing by domain-dependent features such as company names, dates, locations, etc. The last type of indexing is a first step towards integrating detection and extraction systems.

In more detail, the document structure is used to identify which parts will be used for indexing. The first step of this process is then to scan for word tokens. Most types of words (including numbers) are indexed, although a stopword list is used to remove very common words. Stopwords can be indexed, however, if they are capitalized (but not at the start of sentences) or joined with other words (e.g. "the The-1 system"). Words are then stemmed to conflate variants. Although the Porter stemmer was used for the TREC-2 experiments, we have developed a new stemming algorithm that has a number of advantages for operational systems. A number of recognizers written in *flex* are then used to identify objects such as company names and mark their presence in the document using "meta" index terms. A company name such as IBM in the text, for example, will result in a meta term #COMPANY being recorded at that position in the text. The use of these meta terms extends the range of queries that can be specified. This completes the usual processing for document text.

The document indexing process also involves building the compressed inverted files that are necessary for efficient performance with very large databases. Since positional information is stored, overhead rates are typically about 40% of the original database size.

The query processing process involves a series of steps to identify the important concepts and structure describing a user's information need. INQUERY is unique in that it can represent and use complex structured descriptions in a probabilistic framework. Many of the steps in query processing are the same as those done in document indexing. In addition, a part-of-speech tagger is to used to identify candidate search phrases. Domain-dependent features are recognized and meta-terms inserted into the query representation. The relative importance of query concepts is also estimated, and relationships between concepts are suggested based on simple grammar rules. An evaluation of some of the query processing techniques is presented in [1].

INQUERY also has the capability of expanding the query using relationships between concepts found by either using manually specified domain knowledge in the form of a simple thesaurus or by corpus analysis. The WORDFINDER system is a version of INQUERY that retrieves concepts that are related to the query. WORDFINDER is constructed by identifying noun groups in the text and representing them by the words that are closely associated with them (i.e. occur in the same text windows). Concept "documents" are then stored in INQUERY. This technique of query expansion was not tested in TREC-2.

The query evaluation process uses the inverted files and the query represented as an inference net to produce a document ranking. The evaluation involves probabilistic inference based on the operators defined in the INQUERY language. These operators define new concepts and how to calculate the belief in those concepts using linguistic and statistical evidence. We are constantly experimenting with and refining these operators (for example, the operator defining a phrase-based concept) in order to improve retrieval performance.

The relevance feedback process uses information from user evaluations of retrieved documents to modify the original query in detection or routing environments. The INQUERY system, because it can represent structured queries, supports a wide range of learning techniques for query modification [5]. In general, new words and phrases are identified in the sample of relevant documents. These are added to the original query and all the terms in the query are then reweighted. With the amount of relevance information available in TIPSTER, relatively simple automatic techniques appear to produce good levels of effectiveness. We are also investigating the effect of using more limited information and more complex learning techniques, such as neural networks.

3 Query Processing

In order to clarify the query processing done for the TREC and TIPSTER experiments with INQUERY, the following sections give more detailed descriptions.

There are two main kinds of query styles: a natural language query and a keyword or key concept query. For example, the <desc> and <narr> fields of a TIPSTER query represent natural language queries of varying levels of abstraction. The <con>, <title> and <fac> fields represent key concepts in the query. The main difference between the two types of processing is that the key concept query has more controlled information. The phrasing and emphasis are already given and do not have to be conjectured from the language structure. It is valuable to discover how to treat both styles of query, because a good user interface will make it easy for a user to input both styles. For example, a user may enter a prose query and then highlight the important words and phrases in the query in some convenient manner. These highlighted words would then be treated as key concepts in the query processing.

3.1 Prose query processing

Natural language query fields are tagged for syntactic category by a part-of-speech (POS) tagger. Currently we use the tagger developed by Ken Church. We have developed our own POS tagger, and we expect to begin using it in the fall of 1993. There are some pre-tagging and post-tagging "housekeeping" operations, such as removing parentheses. (The current version of INQUERY does not permit parentheses except as part of an operator, and we do not yet make any inferences from the presence of parentheses during the text processing.) Additionally, we change operator phrases to single words in order to simplify later processing. An example of this simplification is replacing the phrase *in order to* with the infinitive particle *to* or replacing *with respect to* with the word *regarding.* The goal of this replacement is to remove phrases which resemble noun phrases syntactically but which are really syntactic operators (e.g., phrasal prepositions) with no substantive content. At this stage, *stop phrases* are also removed.

3.1.1 Noun and adjective phrase capture: orthographic and syntactic clues.

When the text is tagged and the potentially irrelevant material has been removed, syntactically-based noun group capture is performed. Certain kinds of noun phrase patterns are enfolded in a #PHRASE operator:

1. A noun phrase which contains more than one modifying adjective and noun is enclosed in a #PHRASE operator;

2. A head noun with no premodifiers and followed by a prepositional phrase is enclosed in a #PHRASE operator with the head noun of the prepositional phrase;

3.1.2 Constraint capture

All text in the query is searched for constraint expressions. Among these expressions are the words *company*, *not U. S.* or a restriction in the nationality section of the <fac> field to U.S. or other nationality. A restriction to U.S. nationality as the area of interest is implemented by penalizing documents for references to foreign countries. A restriction to other nationalities is implemented by repeating that country as a term. This asymmetry depends on the fact that the document collection is drawn solely from U.S. sources, and therefore the U.S., as the default area of interest, is rarely referred to unless the government or foreign policy implementation is under discussion.

There is some recognition of simple time expressions, such as *since 1984* which are expanded to the set of years which might be intended by the phrase in question.

Countries are recognized as such and are handled so that expressions like *South Africa* are phrased as `#1( south africa )` even when they appear in the middle of a larger group of capitalized words. In addition, proper names such as country names are moved out of the scope of `#PHRASE` operators, since it generally increases the effectiveness of a `#PHRASE` to reduce the number of words in it. Nationality constraints can better be maintained within the scope of the larger and more tolerant `#SUM` operator. For example the phrase

"import ban on South African diamonds"

becomes by stages,

```
#PHRASE (import ban on #SYN (#1 (south african) #1 (south africa)) diamonds)
```

and finally

```
#SUM (#SYN (#1(south african) #1(south africa))
#PHRASE(import ban on diamonds)).
```

3.2 Key concept query processing

Key concept query processing is different from prose query processing since the concept separation provided by the user can presumably be trusted. Instead of using a part-of-speech tagger, we rely on comma delimitation of concepts, and #PHRASE the words found between each pair of delimiters.

Additionally, if any constraints were found anywhere else in the query, e.g., a mention of the word *company* or an exclusionary geographical constraint (e.g., *not USA* or *only USA*), the query will be modified according to these constraints. For example,

only USA ⇒ `#NOT (#FOREIGNCOUNTRY )`

and

not USA ⇒ `#NOT ( #USA ).`

If the word company is found in a query, then a second copy of the key concepts (the <con> field), is produced where each item in the field appears in an unordered window operator with the special concept #COMPANY. For example, if the word *South Africa* appears as a key concept (and *company* appears somewhere in the query), then the preprocessor would produce the term `#UW50( #COMPANY #1( south africa))` which would match any document which had a company name within fifty words of *South Africa*.

4 The TREC Experiments

Four experiments were submitted to the TREC evaluation, two "ad-hoc" and two "routing". In these experiments, we emphasized automatic query processing and automatic feedback algorithms for routing. The following is a summary:

- AdHoc: topics 101-150 against TIPSTER volumes 1 and 2.

  INQ001 Created automatically from TIPSTER topics. Contains phrases. Details of query processing used are described below.

  INQ002 INQ001 queries, modified manually. Modifications restricted to eliminating words and phrases, and adding paragraph-level operators around existing words and phrases. The method for doing this was done somewhat differently than last year's TREC conference, as discussed below.

- Routing: topics 51-100 against TIPSTER volume 3.

  INQ003 Created automatically from TIPSTER topics and relevance judgements from Volumes 1 and 2. Baseline queries (from a previous TIPSTER evaluation) were modified by reweighting and adding single-word terms. The term weighting and selection function used was df.idf, as described in [5]. Only the top 120 relevant documents found by INQUERY were used for feedback, and 30 terms were added to each query.

  INQ004 Formed by combining (using the #SUM operator) INQ001 queries and INQRYP queries (used in TIPSTER 18 month evaluation). The INQRYP queries were produced automatically and then modified manually. Modifications restricted to eliminating words and phrases, and adding paragraph-level operators around existing words and phrases.

Query Type		**Average Precision**		
	5 Docs	**30 Docs**	**100 Docs**	**11-Pt Avg**
INQ001	.62	.57	.49	.36
INQ002	.60 (−2.6%)	.59 (+3.5%)	.51 (+4.1%)	.36 (0%)

Table 1: Results for Adhoc queries

Table 1 gives the results for the adhoc queries. These show that there is little difference in effectiveness between the automatically processed queries and the semi-automatically processed queries. The query processing for the automatically processed queries has been significantly improved as described in the previous section, but there is another effect. Compared to the manual query run in the last TREC conference, paragraph-level concepts were formed in a much more mechanistic way and were constrained by the language of the description and the narrative. In the previous conference, the only constraint was the vocabulary used in the queries, and the user's "world knowledge" was used to group concepts.

This resulted in considerably better retrieval performance. Additional experiments using manually edited queries are discussed in the next section.

Query Type	**Average Precision**			
	5 Docs	**30 Docs**	**100 Docs**	**11-Pt Avg**
INQ003	.64	.56	.45	.35
INQ004	.67 (+3.7%)	.58 (+2.7%)	.45 (0%)	.36 (+2.4%)

Table 2: Results for Routing queries

The routing results show that some improvement is obtained by combining the manual queries with the queries that were automatically modified using relevance feedback techniques. The difference in performance between the two types of queries is considerably less than last year, however. Our own experiments have also shown that no additional gains in performance were obtained by using more than the top 150 documents from the INQUERY output. This is a significant result from a practical viewpoint, since in an operational environment we will not want to rely on having output from other systems or need thousands of relevance judgements before performance improves.

5 Other Experiments

In the TIPSTER 24 month evaluation, which took place soon after the TREC-2 evaluation, we did a number of experiments that complement those done in TREC. In particular, we evaluated paragraph-based retrieval, expansion using an automatically generated thesaurus, feedback techniques that use phrases, and Japanese indexing techniques. In this section, we report some of the most interesting results. The precision figures given here are calculated using the TREC-2 relevance judgements, rather than the TIPSTER judgements.

The first two experiments were with adhoc queries. INQ041 (the numbers are consistent with those used in TIPSTER and other publications) is a run that used a different manually modified version of INQ001. That is, the manual modifications were the same as those done in the first TIPSTER and TREC evaluations, rather than the more restricted modifications done for INQ002. INQ042 is a run that combines INQ041 with INQ001.

Query Type	**Average Precision**			
	5 Docs	**30 Docs**	**100 Docs**	**11-Pt Avg**
INQ041	.68	.60	.50	.36
INQ042	.65 (−4.6%)	.61 (+1.7%)	.51 (+2.0%)	.38 (+5.6%)

Table 3: Results for TIPSTER adhoc queries

These results show that the manually modified queries can achieve significantly better precision at low recall levels. For example, at the 5 document cutoff level, the average precision for INQ041 is 9.7% higher than INQ001. The overall average is the same, however. This is a much smaller difference than was seen in the first TREC and TIPSTER evaluations of the INQUERY system and it indicates that the automatic query processing has improved considerably.

The combination search (INQ042) is slightly worse than INQ041 at the 5 document cutoff level, but overall is better than either the automatic or manual queries on their own. An adhoc search that incorporates automatic paragraph-level matching was also tested in TIPSTER and this resulted in a further 5% improvement.

INQ023 and INQ024 are routing query sets that were created automatically using relevance judgements from volumes 1 and 2. In addition to the single-word terms added in INQ003, 10 phrase-level concepts and 20 paragraph-level concepts were added to the query. A phrase-level concept is a #UW5 two-word pattern that occurs frequently in the relevant documents, and a paragraph-level concept is a #UW50 two-word pattern. The #UWn operator looks for co-occurrence in any order in a text window of size n. The difference between INQ023 and INQ024 is that INQ023 contains the original query terms in addition to terms extracted from relevant documents, whereas INQ024 contains only terms from relevant documents.

Query Type	**Average Precision**			
	5 Docs	**30 Docs**	**100 Docs**	**11-Pt Avg**
INQ023	.67	.60	.47	.38
INQ024	.68 (+1.5%)	.59 (−1.7%)	.46 (−2.2%)	.39 (+2.6%)

Table 4: Results for TIPSTER routing queries

These results show that there is little difference between using the original query or just the relevant documents. This is probably due to the large number of relevance judgements available in this routing experiment. In a relevance feedback situation, where there are far fewer relevant documents, the original query is very important. It is clear that the addition of phrase and paragraph-level structure to the routing has improved performance. The average precision for INQ023 is 8.6% higher than INQ003. Combining these new runs with manually modified routing queries produced further improvements.

6 Summary

The TREC-2 runs, both in the adhoc and routing categories, provided further evidence that manually generated queries are not, in general, superior to automatically processed natural language queries. In the case of routing, in fact, the manual queries are significantly less effective. They do, however, improve the effectiveness of retrieval when used in combination with the automatic queries. This combination of query types has been a theme of the research at the University of Massachusetts and has been established as effective in a number of experiments.

The additional TIPSTER runs showed that learning structure in the form of phrases and paragraph-level co-occurrences is effective for routing. They also showed that learning techniques significantly improve performance (the best routing runs were more than 20%

higher in terms of average precision than the best queries that were not modified using relevance judgements). It is becoming apparent that techniques that may not work well in relevance feedback situations with few identified relevant documents, may be very effective in routing where there are many more relevant documents identified. We are currently doing experiments with different forms of weighting, including the use of identified non-relevant documents.

With regard to improving the performance of adhoc queries, we are continuing to carry out experiments with different ways of estimating the probabilities (or tf.idf weights) needed for the inference net, and with different forms of paragraph-level matching. Finally, as mentioned earlier, we have seen some significant improvements using automatic query expansion based on corpus analysis.

References

[1] J. P. Callan and W.B. Croft. 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, 1993.

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

[3] W. Bruce Croft and Howard R. Turtle. Text retrieval and inference. In P. Jacobs, editor, *Text-Based Intelligent Systems*, pages 127–156. Lawrence Erlbaum, 1992.

[4] Hideo Fujii and W. Bruce Croft. A comparison of indexing techniques for japanese text retrieval. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval*, pages 237–246. ACM, 1993.

[5] David Haines and W. Bruce Croft. Relevance feedback and inference networks. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval*, pages 2–11. ACM, 1993.

[6] H.R. Turtle and W.B. Croft. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3):187–222, 1991.

[7] H.R. Turtle and W.B. Croft. A comparison of text retrieval models. *Computer Journal*, 1992. To appear.