Okapi at TREC–4

S E Robertson   S Walker   M M Beaulieu   M Gatford
A Payne

Centre for Interactive Systems Research
Department of Information Science
City University
Northampton Square
London EC1V 0HB
UK

Advisers: E Michael Keen (University of Wales, Aberystwyth), Karen Sparck Jones (Cambridge University), Peter Willett (University of Sheffield)

1 Introduction

Okapi at TRECs 2 and 3

During TRECs 2 and 3

- the new term-weighting functions were developed and refined as described in [1, Section 3.2]
- a method of runtime passage determination and searching was devised [1, Section 4];
- an inefficient but surprisingly effective way of choosing routing terms was developed [1, Section 6]);
- and a reasonably effective way of automatically expanding ad hoc queries with terms from documents retrieved in a pilot search was developed [1, Section 5].

TREC–4

City have submitted interactive runs in all the previous TRECs, with fairly undistinguished results. This time the main emphasis has been on the development of an entirely new interactive ad hoc search system (Sections 3 and Appendix). On the non-interactive side

routing term selection: there has been further work on methods of selecting routing terms;

manual and automatic ad hoc: the automatic ad hoc was done in more or less the same way as for TREC–3, but in view of the very brief topic statements a few runs were also done with manually edited queries.

2 The system

2.1 The Okapi Basic Search System (BSS)

The BSS, which has been used in all City's TREC experiments, is a set-oriented ranked output system designed primarily for probabilistic–type retrieval of textual material using inverted indexes. There is a family of built-in weighting functions as described in [1, Section 3]. In addition to weighting and ranking facilities it has the usual boolean and quasi-boolean (positional) operations. There are a number of non-standard set operations, some of which were added during the TREC–4 experiments.

- Varieties of AND and NOT operators intended for "limiting": these are used to reduce a ranked set to the subset implied by the operator and the right-hand operand(s), without affecting the weights of the elements of the left-hand operand.
- A unary operator which produces a set from the top-ranking N elements of its operand (example below).

```
FIND SET=<setnum> TOP=<num>
```

- A binary NOT–type operator which, when the right-hand operand R is a constituent of the (ranked) left-hand operand L, produces a new set which is what L would be if R had not been a constituent. This was used in routing term selection to speed up testing the effect of removing a term from a query.

Some enhancements were made to the data processing software. Indexes were simplified, and, to accommodate a single index to the million-plus documents of disks 1–3, redesigned so that an index may be distributed over a number of Unix filesystems.

Weighting functions

The functions available were identical to those used in TREC–3. The only ones used during TREC–4 were varieties of the **BM25** function [1, Section 3.2]

$$\sum_{T\in\mathcal{Q}} w^{(1)}\frac{(k_1+1)tf}{K+tf}\frac{(k_3+1)qtf}{k_3+qtf}+k_2.|\mathcal{Q}|.\frac{avdl-dl}{avdl+dl} \quad (1)$$

where

$\mathcal{Q}$ is a query, containing terms T

$w^{(1)}$ is the Robertson–Sparck Jones weight [3]

$$\log\frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)}$$

of T in $\mathcal{Q}$

N is the number of items (documents) in the collection

n is the number of documents containing the term

R is the number of documents known to be relevant to a specific topic

r is the number of relevant documents containing the term

K is $k_1((1-b)+b.dl/avdl)$

k_1, b, k_2 and k_3 are parameters which depend on the database and possibly on the nature of the topics. For the TREC–4 experiments, typical values of k_1, k_3 and b were 1.0–2.0, 8 and 0.6–0.75 resp., and k_2 was zero throughout.

tf is the frequency of occurrence of the term within a specific document

qtf is the frequency of the term within the topic from which Q was derived

dl is the document length (arbitrary units)

$avdl$ is the average document length.

Passage determination and searching

The same technique was used as for TREC–3 [1, Section 4]. The theoretical minimum passage length was one algorithmically determined paragraph, paragraph length and position information being recorded in an auxiliary paragraph file for each database. Passages consisted of an integral number of paragraphs. The run-time arguments are

p_unit: the minimum number of paragraphs in a passage; passage lengths are a multiple of this, unless they hit the end of the document without reaching a multiple

p_step: the number of paragraphs to advance between passages

p_maxlen: the maximum number of paragraphs in a passage.

For example, if $p\_unit = 4$, $p\_step = 2$ and $p\_maxlen = 8$ and a document is 11 paragraphs long, the following passages are examined: 1–4, 1–8, 1–11, 3–6, 3–10, 3–11, 5–8, 5–11, 7–10, 7–11 and 9–11; if $p\_unit = p\_step = 1$ and $p\_maxlen = \infty$ every run of consecutive paragraphs is examined. It is also possible to set a *passage_avedoclen*, typically somewhat lower than the database's true average document length. In the tables, passage runs are sometimes notated in the form $(\langle p\_unit\rangle, \langle p\_step\rangle, \langle p\_maxlen\rangle[, \langle passage\_avedoclen\rangle])$.

Whatever the passage–argument values the entire document was also treated as a passage. The output from a passage search gave for each document a weight for the entire document and the location and weight of the highest–scoring passage. Obviously, passage searching is computationally demanding for long documents, with time complexity increasing with the square, or cube (if *p_maxlen* is infinite), of the document length, so it is essential to keep the number of documents examined as small as possible. This is achieved by performing passage searching in two stages. In the initial search whole documents only are examined, and the top ten thousand or so output to form an intermediate set, which it is hoped will contain most of the documents which would be found by a 1000–cutoff passage search. In the final stage passage searching is done on the intermediate set only, thus reducing the number of documents examined from perhaps a few hundred thousand to ten thousand.

Passage searching was used in all the official City runs, and usually gave an improvement in average precision of about 2–5 percent, sometimes at a cost in low precision. Most of the runs were done with passage arguments $(1, 1, 20)$ or $(4, 2, 32)$. There is little difference between the results, and the latter is quicker. In the interactive system the passage information enabled the interface to point the user to the "best" part of a long document.

2.2 Hardware

A single-processor Sun SS10 with 256 MB of core and about 25 GB of secondary storage was used as the main development machine and file server. Batch processing was also done on three other Suns, an IPX with 48 MB, a 4/330 with 40 MB and an IPC with almost no memory.

2.3 Databases

Three databases were used: disks 1, 2 & 3 for routing trials and some of the ad hoc development, disks 2 & 3 for automatic, manual and interactive ad hoc, and "disk 4" for the predictive routing runs. The same three–field structure was used, common to all source datasets, as for TREC–3. The first field was always the DOCNO and the third field contained all the searchable text, mainly the TEXT portions but also headline or title-like material for some datasets and documents. The second field was unindexed (and unsearchable) and so only (possibly) useful for display to users of the interactive system. It was empty except in the case of SJM, when it contained the DESCRIPT field; and the Ziff JOURNAL, AUTHOR and DESCRIPTORS fields. All the databases were set up in such a way that search-time passage determination and searching could be done.

3 Interactive track experiment

For TREC–4 the Okapi team concentrated more on the interactive track. The object was to build on our previous experience in both automatic and interactive methods and optimize on a combined approach. The focus was on two aspects: the development of an interface more amenable for interactive query formulation, and on passage retrieval for relevance feedback. The searchers were members of the team who were experienced with the system and played the role of an intermediary searching on behalf of a remote user. A post-search questionnaire was administered in order to obtain more qualitative data on the searchers' perception of the query, the effort required in selecting terms and making relevance judgments, and overall search satisfaction.

The Appendix includes a description of the interactive system, a short summary report on the search process, including a search narrative for query 236 on areas of disagreement on the laws of the sea, as well as tabular results of the questionnaire and the guidelines given to the searchers (E).

A major concern in carrying out the interactive experiment was to ensure that the task could be carried out in the time allocated with minimal loading on the user, whilst at the same time exploiting the system's functionality to the maximum. Unlike the command-based expert interface used in TREC–3, the current GUI interface made full use of direct manipulation interaction for making selections and executing commands and for the concurrent display of different information. The main features and improvements made it easier for the searcher to keep track of the search as it developed and included the following:

- the ability to enter search terms as phrases (interpreted by the system as a combination of an adjacency search and a lower–weight same-sentence search—see Appendix A.2.1);
- the use of a single dynamically ranked and updated current query term set, including query expansion terms (single words) extracted from relevant documents, with interactive addition and removal of terms by the searcher;
- the presentation of a more informative hitlist with short document titles or equivalent, together with source, length and query term occurrence information;
- the direct viewing of highlighted best passages of documents to enable the searcher to make relevance judgments more quickly and also to choose between the full document and best passage only for relevance feedback;
- the display of user selected relevant items in ranked order in a separate window, updating the searcher on the current state of the search results.

A search session could thus be broken down into one or more iterations of three basic sequential activities, each of which was handled by a separate window: entering and manipulating query terms, viewing hitlists of titles and displaying documents to make relevance judgments.

3.1 Generating initial query terms

Since the topics provided so little context, searchers were asked how difficult it was to interpret the topics. Just over half (13) of the topics were found to be straightforward and easy to ascertain what was required. Searchers expressed reservations on nine of the topics, which they described as moderately easy. For the most part they anticipated some difficulty in finding a range of documents to reflect all aspects of the topic. Three queries (205, 209, 211) were deemed to be difficult to formulate because the searcher did not know what sort of documents to expect.

The short topics undoubtedly affected the number of terms used for the initial query formulation. An average of 4.7 terms was used, compared with 8.1 in the interactive routing task for TREC–3. Terms used divided roughly equally between phrases and single words (2.3 and 2.4 respectively). Searchers declared that it was easy or moderately easy to generate initial query terms for 18 of the topics. In the majority of cases this involved both extracting terms directly from the topic and adding other terms. For the six who found it difficult, or in one case very difficult (topic 236), the problem was thinking up synonyms or appropriate words to describe the different aspects of the topic or to express abstract ideas. For three out of the seven queries which were described as difficult or very difficult, only a single term was entered to start the search and the rest had two, three, six and six terms respectively. For the seven queries described as easy, the number of terms ranged from three to nine.

3.2 Expanded query term lists: adding and removing terms

Terms were added to the query during the search either by being entered directly, or as a result of expansion by the system from marked documents. Any terms could be removed by the searchers.

An average of 5.2 terms were added directly by searchers after the initial search and the range was 0–19, with no terms added in six searches. Searchers who added new terms indicated that for 14 of the searches they did so to improve recall, and in only four of the searches was the purpose to shift the emphasis or improve precision. Similarly, searchers seemed to remove terms from term sets primarily to improve recall. In 16 searches proper names were removed and in eight numbers were removed. Terms deemed as inconsequential or irrelevant were discarded in seven searches, whereas in two searches terms were also removed if they appeared both in phrases and as single terms. In only one case a searcher divulged that terms were removed in an attempt to focus the search.

An average of 39.3 terms were removed in the course of a search and the range was 0–113. This far exceeded the number of terms removed in the interactive routing experiment in TREC–3, where the mean was 3.2 and the range 0–15. In the current round a more extensive list of extracted terms was displayed when searchers used the query expansion facility. This list extended beyond the cut-off used by the system when performing a search, in other words, the displayed list included terms lower down the ranking, which would not be used in the search unless brought up by searcher action. Clearly it was easier for users to discard terms suggested by the system than to generate their own. In some instances searchers did find relevant terms in documents and added them to the query term set themselves. Overall searchers depended heavily on the query expansion facility to formulate their queries.

3.3 Query expansion and final query terms

Searchers arrived at a final query formulation normally after an average of 3.6 iterations. An iteration here was defined as the execution of the search command, which may or may not have been preceded by the use of the query expansion facility. Query expansion was used only once in 20 of the searches, twice in two searches and not at all in three. For 11 searches where expansion took place, searchers indicated that they had expanded because they had found relevant documents but had exhausted the current list. Their intention was to get more items similar to what had already been retrieved. In five cases the intention was to find documents which were different from those already found to be relevant and in six searches query expansion was used out of desperation because not enough relevant documents could be found.

The length of the topics had little impact on the number of terms in the final query where the mean was 16.9 and 16.6 for TREC–3 and TREC–4 respectively. The main explanation for this is simply that for interactive searching the system used the 20 top ranking terms for query expansion and the searcher had the option to remove some. Out of the average number of 16.6 terms, 12.8 were individual terms and 3.8 were phrases. Overall the system retained 121 out of the total of 133 phrases generated by the searchers.

Searchers did not find it difficult to determine what terms should constitute the offline query for the secondary task. For 21 searches they found it easy or moderately easy and for only four they found it difficult or very difficult. In thirteen searches the final interactive query formulation was altered for the secondary offline stage of the interactive task. In 11 of the searches, terms were both added and removed, whereas in the remaining two searches terms were added or removed only.

3.4 Viewing hitlists of titles

In just over half of the searches one or two hitlists of ranked items were displayed, and between three and seven hitlists in the remainder. On average searchers browsed through 54.2 titles per search. Searchers were asked how easy it was to find relevant items in the initial hitlist and in 13 instances they found it easy or moderately easy compared to 12 where they found it difficult or very difficult. There appeared to be a strong correlation between users' assessment of the degree of ease in generating initial query terms and their perception of the quality of the results in the initial hitlist. For example it turned out to be difficult or very difficult to identify possible relevant items from the hitlist for five out of the six searches where searchers experienced problems in generating appropriate query terms in the first place.

3.5 Passage retrieval, relevance judgments and relevance feedback

For the interactive experiment passage retrieval was applied after the initial retrieval of the document set and served to re-rank the documents according to best passage. Records were displayed with the best passage and query terms both highlighted. In some cases the weighting operation did not identify a best passage [1] and records were displayed with query terms highlighted only. In the case of very long documents (more than 10K), only the best passage was displayed.

The purpose of passage retrieval in interactive searching was to:

- improve the ranking of documents
- enhance the display and viewing of documents
- assist users in making relevance judgments
- improve term selection for query expansion.

By highlighting the section of the document where query terms were the most concentrated, the searcher could use this as a starting point. This was particularly helpful for long documents which might have required more extensive scanning. In the case of documents dealing with several topics, best passage retrieval made it possible to go straight to the section most appropriate to the query.

In making relevance judgments, searchers were in effect not only indicating the relevance of the document for the query but they were also choosing between the whole document or the passage as suggested by the system for extracting terms for query expansion. This was particularly useful in the case of multi-topic documents. For 64% of positive relevance judgments, the searcher indicated that the system–specified best passage was relevant (this implied that *only* this passage was used for query expansion). On average searchers made 17 positive relevance judgments per search as opposed to 14 negative ones. Diagnostics on the effectiveness of passage retrieval for interactive searching have yet to be carried out to compare the ranking of documents, the degree of agreement between the system and the user for term selection as well as the retrieval effectiveness of using whole documents or best passages only for query expansion.

3.6 User satisfaction

An attempt was made to get some measure of user satisfaction by asking searchers how they rated the overall difficulty of the topics and how they perceived the success of their searches. Fifteen topics were rated as easy or moderately easy and ten as difficult or very difficult. Searches were classed as problematic when searchers found it difficult to express the search more precisely and to focus on a particular aspect.

With respect to how successful they considered their searches, nine were considered to be successful, another nine moderately successful and seven not successful. Success seemed to be expressed here both in terms of recall and precision. Unsuccessful searches seemed to be those where few items were found and the searcher felt that perhaps there should have been more there. In the case of moderately successful searches, these seemed to be searches where the relevant items covered some aspects of the topic but not all. Six out of the seven searches categorized as unsuccessful were in fact deemed to have produced poor results in the initial hitlist. Of the remaining six searches which also produced poor initial results, four turned out to be moderately successful and two successful.

3.7 Results of the interactive searches

For the primary task (Table 8 in the Appendix), the macro–average recall and precision figures are 15% and 66% respectively. An average of 0.38 relevant documents were found per minute of elapsed search time.

Further measures were also calculated based on different time intervals and the number of iterations per search sessions (Tables 9 and 10 in the Appendix). Taking into account the end point results in these tables, searchers saw on average 31 full documents per session (column K), that is one per minute, and made positive relevant judgments for just over half of these (17, column F). Of the 14.5 average number of official relevant documents seen by the searcher (column L), 3 (column M) or 24% (column N) were rejected. Figure 1a shows that as more documents were seen over time in a search session, more positive relevance judgments were made, an indication that the search formulation improved. Also of those items selected by the searcher (Figure 1c, H), the proportion officially judged relevant improved in the course of the search, showing that searchers became more confident and familiar with the topic. This improvement happens in the first 10 minutes.

The same measures calculated at each iteration demonstrate the same trend. An iteration is defined here as a submission of a new search statement. The flattening effect in Figures 1b and 1d is partly due to the fact that not all the searches reached the maximum number of iterations. This does not occur in the case of the duration of search sessions because no session ended before the penultimate 20 minutes point. However, the flattening effect on precision appears to be genuine (Figure 1c). Clearly the initial queries are not very good but improve as they are reformulated.

For the secondary task, the results are given in Table 1 below, together with our automatic and manual ad hoc results, as well as a diagnostic run. The diagnostic run was based on the final query formulation without the assumption that some documents had been retrieved (i.e. without frozen ranks). The very high precision at 5 documents reflects that this is an "optimal" formulation, whereas the results of the actual interactive session

[1] i.e. the best passage was in fact the whole document.

Table 1: Secondary task evaluation summary compared with City non-interactive ad hoc results

Run	AveP	P5	P30	P100	R-Prec	Rcl
Cityi1	0.274	0.520	0.456	0.321	0.340	0.543
Citya1	0.243	0.480	0.353	0.235	0.279	0.531
Citym1	0.235	0.464	0.335	0.233	0.272	0.531
Citydiag	0.241	0.592	0.425	0.285	0.285	0.530

reflect searchers' initial uncertainty. The performance of the diagnostic run seems to be worse at high document ranks, so that the average precision comes down to more or less the same as the automatic ad hoc results (citya1).

4 Automatic routing

The task involves selecting terms, assigning weights to them and combining the terms in a suitable way, while hoping that whatever training datasets are used are not too different from the test data. One could in theory take all the index terms in some training set and perform some multidimensional combinatorial feat to arrive at an approach to an optimal set of (term, weight) pairs. In practice, none of us has enough computing resources even to reach some state of sub-optimality; enough constraints have to be added to reduce the task to something which can be done in a matter of a few weeks.

The constraints used were as follows:

1. terms were restricted to those present in officially relevant documents;
2. terms were assigned $w^{(1)}$ weights (Section 2.1) in accordance with their occurrence in the relevant documents;
3. terms were arranged in descending order of their Robertson Selection Value (RSV) [4];
4. the number of terms considered by the second–level selection process was limited to a fixed number, never greater than 200;
5. terms were added to or removed from the query singly, so that only a minute proportion of the possible combinations of terms were considered.

This type of procedure was first used by City for TREC–3, with encouraging results; for TREC–1 and TREC–2 [5] first–level term selection was as above (items 1–3), but there was no second–level selection—either a fixed number of terms was used working down the list, or the number of terms used depended on the topic.

4.1 Term extraction

Three sets of potential query terms were produced, one from the full disks 1, 2 & 3 database (*full*) and one each from its odd and even half-collection sub-databases (*odd* and *even* resp). Every non-stop term was extracted and $w^{(1)}$ weights, RSVs and other statistics recorded. Those with RSV less than 3 were discarded.

4.2 Scoring functions for term selection

In TREC–3 the second–level term selection score was in most cases the average precision (using cutoff at 1000 documents) from the official TREC evaluation procedure. At every stage a new search process was invoked, the IDs of the top 1000 documents output, and the TREC evaluation run in the usual way. This rendered the process so slow that a number of constraints were used simply to enable the selection to run acceptably fast: the total number of terms was limited to 20, no term was reconsidered if it had failed to increase the score at some previous stage, the process stopped if eight successive terms failed and, finally, an absolute time limit was applied.

For TREC–4 a number of more rapidly computable measures were tried.

BROOKES The "Brookes measure" [2], is a normalized form of the difference in mean weights between relevant and non-relevant documents in the retrieved set (see equation 2).

$$\frac{\mu_{rel} - \mu_{nonrel}}{\sqrt{(\sigma^2_{rel} + \sigma^2_{nonrel})}} \qquad (2)$$

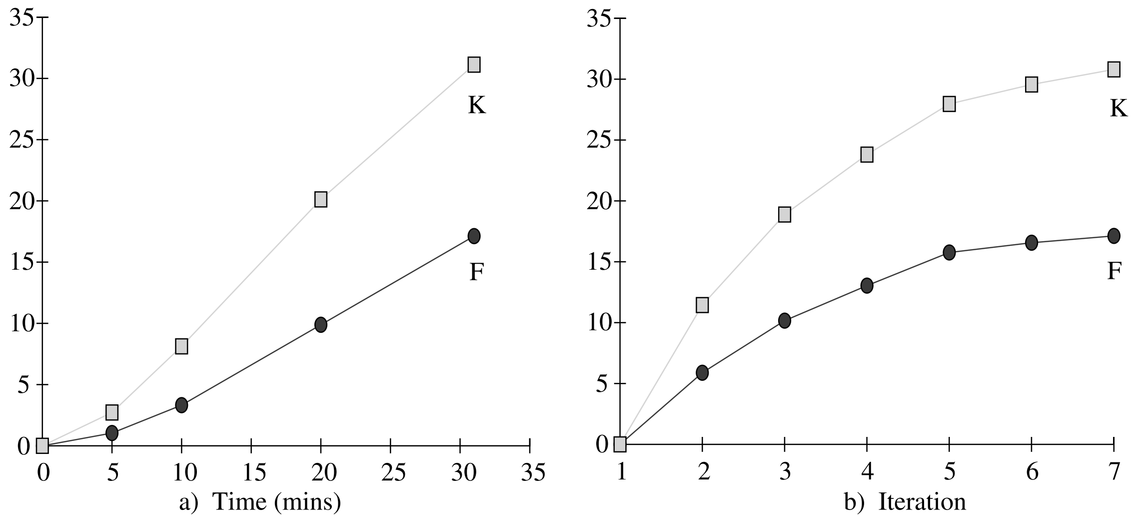
This was efficiently evaluated by calculating the sum and sum of squares of document scores on the fly within the BSS set combination routines. The Brookes measure, although intuitively appealing, did not prove to correlate well with the TREC average precision measure.

DIFFM Un-normalized difference of means between relevant and non-relevant documents. This gave results which were very similar to those from the Brookes measure.

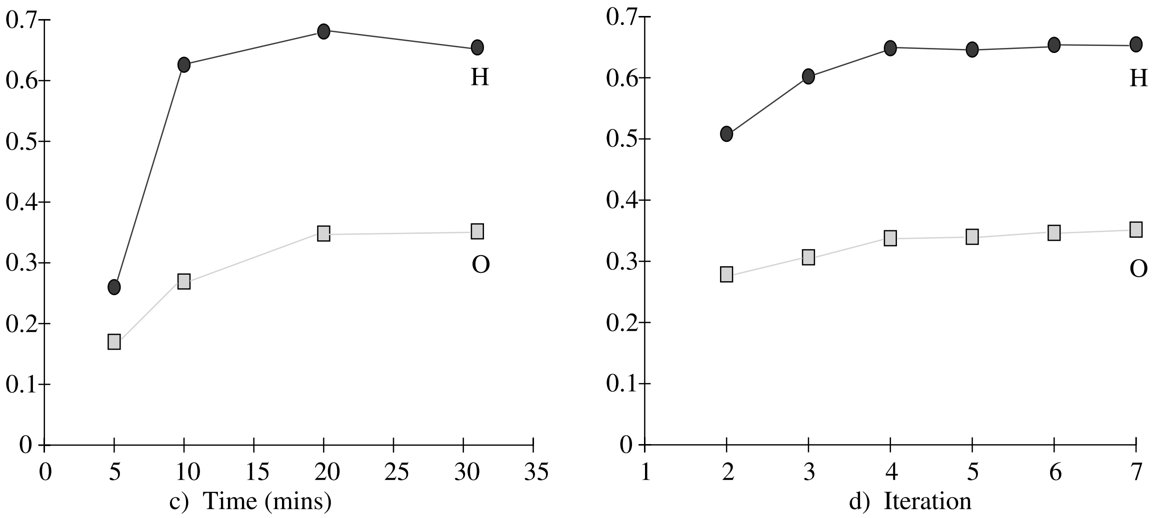
3–PT A 3–point precision average—the mean of precisions at $2R$, R and $R/2$.

RPREC The precision after R documents have been retrieved.

AVEP A multi-point precision average on the top D documents, usually calculated on $D = 1000$ and at D/g, $2D/g$, $\ldots$, D, where the granularity parameter g was varied between 3 and 20. This gave the best results, noticeably better than our TREC–3 results using TREC evaluation average precision



F: Number of searchers' relevance judgements
K: Number of full records seen by searchers



H: Precision of selected items: number of official relevant chosen by the searcher as a proportion of total number chosen by the searcher.
O: System precision: number of items chosen by the searcher as a proportion of total seen by the searcher.

Figure 1: Searcher relevance judgments, Documents shown against Time, Iteration boundaries

under restrictive conditions on the number of terms tried, etc.

TREC–AVEP Average precision identical to that produced by the official TREC evaluation program.

CBM1 This is a linear combination of *TREC–AVEP* and *BROOKES* with *TREC–AVEP* being given ten times the weight of *BROOKES* since their mean values are roughly in this ratio.

The effect of varying scoring function is shown in Table 2

4.3 Term selection algorithms

For all selection methods, potential terms were first arranged in descending *RSV* order to form a term-set $\mathcal{T}$ containing a predetermined maximum number T of terms. For almost all runs terms which occurred more than once in the topic statement were given a bonus by using a k_3 value of 8 in the weighting formula 1. When an iteration starts there is assumed to be a set of already selected terms (possibly empty) forming the current query $\mathcal{Q}$. Almost always, terms in $\mathcal{T}$ were tried in descending order; in some trials a number of terms from the top of the pool were preselected to form an initial current query; in a few trials the top N terms were randomized to test the effectiveness of the *RSV* ordering.

In general the selection procedures involved, for each term T in $\mathcal{T}$, trying the effect of adding T to $\mathcal{Q}$ if it was not in $\mathcal{Q}$; or removing T if it was already in $\mathcal{Q}$. Some of the trial runs used addition of terms only—once a term was in $\mathcal{Q}$ it remained for ever; but it was soon noticed that later removal was sometimes beneficial. To speed the procedures, terms were usually no longer tested after their addition or removal had failed to increase the score on more than some predetermined number (usually four) of successive iterations; in a few trials these "dropped" terms were re-examined occasionally, but this appears to be of only small benefit.

1 *Find best* **(***FB***).** In each iteration every undropped term was examined. At the end of the iteration the highest scoring term was added to or removed from $\mathcal{Q}$. Variants included *removing only* (when the initial query contained all the terms in $\mathcal{T}$); and *adding only*, when the initial query was empty or contained only a few terms.

2 *Choose first positive (CFP).* In each iteration the first term which increases the score by an amount greater than a predetermined threshold percentage is added to or removed from $\mathcal{Q}$. Obviously this goes more quickly than *FB*, but might be expected to give results which are less good. In fact, it appears to give slightly better results.

3 *Choose all positive (CAP).* Every term is considered and immediately added to or removed from the query if it increases the score by more than the threshold. An iteration ends when all non-dropped terms have been considered. Runtime is comparable with method *CFP* (fewer but much longer iterations).

A number of stopping criteria were built in, including elapsed time, maximum number of iterations and maximum query size, but in practice almost all runs ended at the point where there was no single term whose addition or removal increased the score.

4.4 Term selection databases

At least three databases may be used: one for the extraction of terms from relevant documents and their weighting and ranking, one for term selection and one for comparative evaluation of the queries. On the one hand, one needs to use as much of the available relevance information as possible, suggesting that the full database should be used for everything; but there is then a danger of over-fitting (a query consisting of a handful of rare terms from each of the known relevant documents might score very highly on its source database but poorly on any other). Probably there is a need for compromise, or the procedure used for an individual topic should depend on the amount of relevance information for that topic.

In choosing selection algorithms and scoring function for the official runs, we used the odd database both as the source for the term pool and for term selection, but the even database for comparison of methods. That is, the queries were entirely produced from the odd database but then evaluated on the even; the evaluation database thus being independent of the one used to derive the queries, although obviously very similar in makeup.

However, having chosen selection and scoring methods, the official queries were produced using the whole of the training database (as was done in TREC–3). Since the TREC–4 conference some inconclusive experiments have been done in an attempt to compare the effect of using various database combinations. Table 4 may suggest that, at least where there is a relatively large amount of relevance information, there is little difference between the four database combinations tried.

4.5 Routing experiments and discussion of results

In a general routing situation the object is, given a training database containing known relevant and non-relevant documents, to find good methods of weighting and ordering the terms, finding suitable term pool sizes, scoring functions which are good predictors of results on the test database (which, it must be assumed, will be reasonably similar in makeup to the training database), and good algorithms for term selection from the pool. In our case, we disregarded the question of weighting and term ordering. The weighting follows the Robertson/Sparck Jones theory [3], and term ordering was always by Robertson Selection Value [4] (but term ordering affects some of the selection algorithms more than others).

This still leaves a very large number of combinations to explore and selection runs can be slow. A recent estimate gave a very rough figure of 500 cpu–days on a fastish machine for a rather minimal, even preliminary, exploration of the space. It follows that the pre-deadline decisions were based on hope and pants-seats as much as on hard evidence.[2]

Most of the trials reported in this section were done after the TREC–4 relevance judgments were available, using a subset (*T22*) of 22 of the topics[3] derived by selecting alternate topics and then eliminating three for which there were very few relevant documents in the test database. For the most part, the odd database (together with the topic statements and the "odd" relevant documents) was used to derive, weight and select terms; and the even database was used for evaluation. Evaluation was then checked using the test database. All evaluation runs were done at cutoff 1000 documents.

A t-statistic for the difference of mean score (i.e. average precision) between pairs of runs was calculated, the observations being the difference in score for each topic of the 22 sample topics. The validity of this procedure may be questioned, but it gives some indication of the likely significance of a result. It is noticeable that the between-topic variation is much higher than most between-treatment variation. The official results (Table 5) bear this out in the difference between the Cityr1 and Cityr2 runs: the former used the same procedures for all topics whereas for Cityr2 the best method was chosen for each topic. (There is also a large between-database component.) A full analysis of variance would need a very large amount of data to be useful.

Scoring functions

The initial object was to improve on the very inefficient and hence constrained procedure used in TREC–3, in which TREC average precision scores were used but with severe constraints on the number of terms considered and the number of successive "failures", no re-evaluation of a term after it has once failed to increase the score, and limits on runtime. The Brookes score was rapidly calculable, and it was hoped that it would correlate fairly well with some at least of the official TREC scoring measures. A few trials were done using the TREC–3 method to act as a baseline. It rapidly became clear that the Brookes score, at least when applied to a set with no cutoff, was a very poor predictor. It was less bad when evaluated on the top 1000 documents, or a small multiple of R, than on complete sets, but still gave poorer results than the TREC–3 procedure.

With hindsight it is obvious that at this stage we should have simply written a reasonably efficient procedure for calculating TREC average precision on the fly, but this was not done until after the TREC conference. Instead, a number of compromise measures were tried. In an attempt to make *BROOKES* more sensitive to the top end, the weights used in calculating the measure were raised to powers greater then one, but this made little difference. The un-normalized difference of means was tried: this gave results not significantly worse than *BROOKES*. The precision at R documents (*RPREC*) was tried, as it had been for TREC–3, but *RPREC* is not a good predictor of average precision; TREC–3 experience showed that average precision scoring resulted in higher values of *RPREC* than *RPREC* scoring. It also tended to get "stuck", with no single term's addition or removal giving an increase in score. Various types of precision averaging were then tried, taken over up to 20 points. Not surprisingly, these gave results which were nearer to TREC average precision, and one of them (granularity 20) was used for the official Cityr1 queries.

Finally, after the conference, real TREC average precisions were used, sometimes in linear combinations with *BROOKES* (*CBM1*). Some recent results are summarized in Table 2. On 1000 documents, *BROOKES* turns out not significantly inferior to *TREC–AVEP* when evaluated on the test database. However, it is greatly inferior on the training database (difference significant at 0.995), comparable with *RPREC*; it is not obvious how to account for this. Further trials are needed on larger term pools and more topics. The linear combination *CBM1* appears to be very similar to *TREC–AVEP*.

Selection algorithms

A summary comparison of selection algorithms is given in Table 3. It was expected that adding the "best" term at each iteration would give the best results, but this does not appear to be the case. There is surprisingly little difference between the methods, although when done on all the topics *CAP* is confirmed to be the best. It is something of a surprise that *FB* looks to be the worst; it is, unlike the other methods, independent of term order-

[2] Runtimes on an SS10 on 25 topics and a pool of 50 terms vary between about four and 15 hours depending on scoring and selection method; 50 topics and 200 terms can take up to a week. (Obtaining the term pools also takes many hours, but of course this only has to be done once.)

[3] 3, 14, 20, 28, 30, 32, 36, 41, 44, 46, 48, 66, 82, 94, 96, 103, 113, 117, 123, 142, 161, 174

Table 2: Automatic routing: comparison of term selection scoring functions

Scoring function	Mean query len	Score even db	test db	% increase
Baseline run: topic terms, no relevance information	30	0.279	0.288	0.0
topic terms from term pool	21	0.294	0.318	10.4
top 24 terms from term pool	24	0.266	0.343	19.1
top 50 terms	50	0.248	0.316	9.7
top 100 terms	100	0.221	0.286	-0.7
TREC–AVEP	24	0.378	0.402	39.6
AVEP, granularity 10	23	0.372	0.397	37.8
AVEP, granularity 3		0.359	0.389	35.1
BROOKES	22	0.347	0.394	36.8
DIFFM	23	0.340	0.379	31.6
3–PT	17	0.359	0.356	23.6
RPREC	12	0.342	0.340	18.1
CBM1	25	0.374	0.401	39.2

Topics: *T22*.
Terms from odd database; term pool 50 & selection algorithm *CAP* unless stated.
Selection on odd database and assessment on even and test databases.
Cutoff 1000 throughout. Scores are official TREC average precisions.

ing (within the term pool), and one would expect this to be beneficial. When observed—all these runs were logged in great detail—it seems to have something of a tendency to get stuck, easily reaching a stage where no single term gives any increase in score. The first and second *FB* rows in Table 3 are not significantly different from the first *CAP* row on the evaluation (even) database, but all three *FB* rows are significantly worse ($P = 0.9$) on the test database. *CFP* seems to lie between the other two methods. Retrying "failed" terms after a few iterations is occasionally beneficial. Disallowing term removal has a small but noticeable detrimental effect.

Official results

Table 5 gives some test database results on the whole topic set, including the official Cityr1 and Cityr2 runs. While City did fairly well again, there is still clearly a lot of scope for improvement. Both the runs are better (relative to other good runs) at the low-recall end than the high end. Both runs show average precision greater than or equal to the median of all the comparable runs on 44 out of 50 topics, but Cityr2 returned 49:1 on the median precision at 100 documents (and the "1" was topic 50, which had very few relevant documents).

Summary and conclusions

TREC–AVEP appears to be the best scoring function, perhaps not surprisingly as the evaluation was done on that measure. That aside, *TREC–AVEP* is quite a good predictor for other evaluation measures such as R-precision (e.g. one is likely to get higher mean R-precision by optimizing on average precision than on R-precision itself). The simple Brookes measure correlates only weakly with average precision, and gave poor evaluation results on the even database; hence it wasn't even considered for the official runs. However, when evaluated on the test database, results were not much inferior to *TREC–AVEP*, and a linear combination of the two seems to be about as good as the latter. This presumably has something to do with the large disparity between the two databases, and it seems doubtful whether *BROOKES* would be a good choice in a real-life routing situation.

As regards selection algorithm, *CAP* is probably the best, though not by very much. This is a surprising result which needs explaining. It suggests that the ordering of terms by *RSV* is of benefit. It doesn't do so well when the order of the terms in the pool is randomized. Varying the size of the initial set between zero and 50 terms didn't make much difference; sometimes, though, a non-empty initial set containing some of the "better" terms seems to render the selection procedure less likely to get prematurely "stuck". Increasing the number of terms in the pool was beneficial. Again, this may have been partly because it rendered the selection algorithm less likely to stick—it was noticeable that sometimes very low-ranking terms would be added at an early stage, only to be later removed.

On the question of databases, it has not been possible to reach even tentative conclusions. Retrospectively, the benefit of using all the relevance information appears to more than compensate for the dubious use of

Table 3: Automatic routing: effect of varying the selection method

Scoring method	Mean query len	Score on odd	Score on even	Score on test
CAP	24	0.449	0.378	0.402
CAP, add only	28	0.436	0.372	0.395
FB	20	0.438	0.370	0.390
FB with retry	20	0.441	0.371	0.390
FB, add only	19	0.428	0.370	0.385
CFP	25	0.444	0.373	0.392
CFP with retry	22	0.446	0.373	0.400
CFP, add only	27	0.429	0.370	0.393

Topics: *T22*.
Terms from odd database; term pool 50.
Selection on odd database. *TREC–AVEP* scoring.
Assessment on even and test databases.
Scores are official TREC average precisions.
Cutoff 1000 throughout.

Table 4: Automatic routing: effect of varying term source and selection databases

Terms from	Selection on	Mean query len	Score on full	Score on odd	Score on even	Score on test
full	full	24	0.397			0.402
full	odd	23		0.447	0.387	0.400
odd	full	23	0.389			0.396
odd	even	22	0.377		0.432	0.395

Topics: *T22*.
Term pool 50. *TREC–AVEP* scoring. *CAP* selection.
The scores are official TREC average precisions.
Cutoff 1000 throughout.

Table 5: Automatic routing: predictive results on test database, terms & selection on full training database

Scoring function	Selection algorithm	$\|T\|$	Initial query	Notes	AveP	P5	P30	P100	R-Prec	Rcl
AVEP	varies	varies	varies	passages(4, 2, 32)	0.407	0.688	0.571	0.465	0.422	0.844
As row above, but no passages					0.390	0.688	0.573	0.442	0.406	0.832
AVEP	*CAP*	200	3	passages(1, 1, 20)	0.394	0.678	0.558	0.435	0.405	0.860
CBM1	*CAP*	200	3	no passages	0.390	0.668	0.583	0.443	0.413	0.825
AVEP	*CFP*	150	3	passages	0.389	0.700	0.559	0.440	0.405	0.834
AVEP	*CAP*	100	3	passages	0.387	0.684	0.569	0.444	0.408	0.829
As row above, but no passages					0.373	0.700	0.571	0.435	0.398	0.807
AVEP	*CAP*	150	3	no passages	0.378	0.720	0.577	0.434	0.401	0.811
AVEP	*CB*	100	3	no passages	0.319	0.640	0.530	0.385	0.356	0.751

The first row is the official Cityr2 and the sixth is Cityr1.

the same database both as term source and for term selection. Significantly the best test database results were obtained in this way. But using the same database is probably not a satisfactory way of obtaining evidence about the relative merits of the various scoring functions and selection procedures.

5 Non-interactive ad hoc

In TREC–3 City had some success with deriving the ad hoc queries from terms extracted from the top-ranked documents retrieved by a pilot search on the topic terms [1, Section 5]. It was not obvious that this technique, which must depend on reasonably high precision at the low end, would work with the very short TREC–4 ad hoc topics. It turned out, though, that such query expansion was still beneficial, leading to an increase of 20% or more in average precision (with perhaps a small decrease in mean low–end precision). Initial trials showed that results were still very poor compared with those obtained using the fuller topic statements, so it was decided to try the effect of some manual editing of the queries, both before and after expansion. The editing was done without any trial searches. The (unwritten) procedure was something like "Read the topic statement, then remove any term you think is likely to be detrimental in a search for documents about this topic; don't spend too long thinking about it".

Initial queries

Most of the trial runs were done using the only the DESCRIPTION fields of the TREC–4 routing topics on the disks 2 & 3 database (the live runs, of course, used the full [!] topics 202–250). In some runs, passage searching was used. For the manual runs, one of the team members pre-edited the topic statements by removing terms thought to be detrimental; no terms were added. For example, topic 207

> What are the prospects of the Quebec separatists achieving independence from the rest of Canada?

became

> Quebec separatists achieving independence Canada

In effect, "prospects" and "rest" were removed, because the other words would have been stopped in any case.

Query expansion

The top R documents from the pilot search were output and all terms other than stop and semi-stop terms extracted. These were $w^{(1)}$–weighted in accordance with their occurrence in the top documents, with topic terms loaded on the basis of their having occurred in r' out of R' fictional relevant documents (usually 19 out of 20, although more extreme loadings were also tried). The terms were then arranged by descending RSV value, and the required number taken from the top of the list, any term with RSV less than 3 being discarded.[4] For manually post-edited queries one of the team members then removed unwanted terms.

Results

A selection of trial results is in Table 6[5] and final results in Table 7.

It is clear that expansion was still beneficial despite the brief topics—compare the expansion runs in Table 6 with the first control row, in which no expansion was used. However, no expansion compensated for the absence of topic TITLE and CONCEPTS fields. Not surprisingly, the mean low precision is worse in the expansion runs, although it varies widely between topics. Trial results were rather flat over 20–100 documents and 15–30 terms. There is little or no evidence that the use of passages in the pilot search gave any improvement, nor that the topic–term loading was useful.

With regard to the manually edited queries, pre-editing gave a considerable improvement in the trial situation, but post-editing seems to have had a neutral effect in the trials and a detrimental one when used on topics 202–250. More investigation is needed, although it would be hard to obtain objective comparisons across different topic sets and editors.

6 Conclusions

The interactive track at TREC is beginning to bear fruit. What we need now is extensive diagnostic analyses, to fill out the evidence provided by the summary results.

The routing task continues to show that relatively heavy computation based on the training set can produce good results. More detailed conclusions about these methods are given in section 4.5 above.

References

[1] Robertson S E *et al.* Okapi at TREC–3. In: [6]. p109–126.

[4] In a few cases this resulted in the final query containing less than the specified number of terms.

[5] Note that the average precision, R–precision and recall figures in Table 6 are artificially low because we used the full set of disks 1, 2 & 3 relevance judgments instead of disks 2 & 3 only.

Table 6: Non-interactive ad hoc trials

Feedback docs	Feedback terms	Notes	AveP	P5	P30	P100	R-Prec	Rcl
Control runs								
0	0	no expansion, no passages	0.062	0.392	0.299	0.234	0.145	0.225
0	0	no expansion, no passages, topic fields TCND	0.119	0.492	0.414	0.347	0.221	0.349
50	20	manual pre-editing, final passages	0.100	0.396	0.350	0.295	0.194	0.309
50	≤ 20	manual pre- & post-editing, final passages	0.099	0.412	0.363	0.304	0.191	0.307
50	20	initial & final passages	0.092	0.312	0.289	0.277	0.187	0.301
50	20	final passages	0.089	0.356	0.321	0.273	0.178	0.286
50	20	no passages	0.087	0.352	0.318	0.274	0.176	0.282
50	20	no passages, no loading	0.086	0.384	0.316	0.269	0.175	0.280
50	15	final passages	0.090	0.332	0.316	0.275	0.182	0.287
30	20	no passages	0.086	0.336	0.313	0.272	0.176	0.280
30	20	no passages, no loading	0.086	0.348	0.311	0.272	0.175	0.281
50	30	no passages, no loading	0.086	0.364	0.314	0.270	0.173	0.281
50	10	no passages	0.085	0.336	0.312	0.266	0.175	0.269
100	20	no passages, no loading	0.081	0.324	0.283	0.255	0.166	0.270
100	30	no passages, no loading	0.079	0.320	0.280	0.255	0.163	0.264
30	50	no passages	0.080	0.300	0.294	0.261	0.171	0.274
DB disks 2 & 3, TREC–4 routing topics, topic DESC field only & 19/20 loading except where stated								

Table 7: Non-interactive ad hoc results

Feedback docs	Feedback terms	Notes	AveP	P5	P30	P100	R-Prec	Rcl
50	20	initial & final passages	0.257	0.522	0.394	0.274	0.298	0.586
50	20	initial passages	0.250	0.535	0.389	0.271	0.298	0.581
50	≤ 30	manual pre- & post-edit, final passages	0.226	0.498	0.368	0.254	0.271	0.544
0	0	final passages, no expansion	0.209	0.502	0.342	0.235	0.265	0.531
0	0	no passages, no expansion, manual pre-edit	0.215	0.518	0.350	0.242	0.279	0.536
0	0	no passages, no expansion	0.207	0.526	0.349	0.238	0.276	0.529
The top row in the official Citya1 and the third row Citym1 DB disks 2 & 3, topics 202–250, 19/20 topic term loading								

[2] Brookes B C. The measure of information retrieval effectiveness proposed by Swets. *Journal of Documentation 24* 1968 p41–54.

[3] Robertson S E and Sparck Jones K. Relevance weighting of search terms. *Journal of the American Society for Information Science 27* May–June 1976 p129–146.

[4] Robertson S E. On term selection for query expansion. *Journal of Documentation 46* Dec 1990 p359–364.

[5] Robertson S E *et al.* Okapi at TREC–2. In: [7]. p21–34.

[6] *Overview of the Third Text REtrieval Conference (TREC–3)*. Edited by D K Harman. Gaithersburg, MD: NIST, April 1995.

[7] *The Second Text REtrieval Conference (TREC–2)*. Edited by D K Harman. Gaithersburg, MD: NIST, 1994.

A Interactive system description

A.1 GUI interface

The interface was a GUI to the BSS written in C, C++ and TCL/TK. Figures 2 and 3 show screen dumps of the running system.

The Interface was composed of five main parts.

A.1.1 A query entry box

A.1.2 A query terms display box

A scrollable listbox in which were displayed the ranked list of current query terms.

A.1.3 A hitlist display box

A scrollable listbox in which were displayed the ranked hitlist for a given search.

A.1.4 A relevance judgments display box

A scrollable listbox in which were displayed the ranked list of documents judged as relevant by the user.

Items A.1.1 to A.1.4 were all displayed in one window together with three context-sensitive buttons to:

- SEARCH

  Search on the current query iteration—only selectable once a query had been defined.

- EXPAND

  Expand the current query iteration—only selectable once positive relevance judgments had been made.

- END SEARCH

  Selectable at any stage in the search.

A.1.5 A full record display box

A pop-up text window in which full records were displayed. At the bottom of this window were three buttons for making relevance judgments. These are described in more detail in Section A.2.4.

A.2 User interaction

A.2.1 User input of query terms

Users entered one or more terms, optionally followed by an adjacency operator as the last non-space character in the line, into the query entry box.

- No adjacency operator. Each term was treated as a single member of the query.

- An adjacency operator. The set of terms were treated, in input order, as both a phrase (ADJ), possibly including intervening stopwords, and as 'within the same sentence' (SAMES). For example, the input line "laws of the sea" would find documents that included, among others, both "laws of the sea" and "Soviet law does not provide for the right of innocent passage in Black Sea waters". Any set of terms input with an adjacency operator will be referred to as a 'phrase'. Internally two sets were generated, S(A) and S(S), with number of postings and weights N(A), W(A) and N(S), W(S) respectively. These were then combined into one "unit" according to the rules:

Sets generated	Use
1. N(A) = N(S) = 0	discard both
2. N(A) = 0, N(S) > 0	S(S), N(S), W(S)
3. N(A) = N(S), N(A) > 0	S(A), N(A), W(A)
4. N(A) > 0 and N(A) < N(S)	S(A), N(A), W(A) and S(S)-S(A), N(S)-N(A), W(S). Count as one term only.

  The weight calculated for each term—user terms, user 'phrases', and system generated terms—was a Robertson/Sparck Jones F4 predictive weight, with halves. In addition, user entered terms were given a loading; little_r and big_R increased by 4 and 5 respectively.

  Terms were displayed in the query listbox in descending order of Robertson Selection value.

A.2.2 Searching

The top N terms from the current query (N <= 20) were combined in a best match operation (bm25). Passage retrieval (Section 2.1) was applied to the document set generated with parameters p_unit = 4, p_step = 2, k1 = 1.6 and b = 0.7. The weight of the document was taken to be the higher of the weight of the full record or the best passage.

A.2.3 Hitlist generation

For each search, the hitlist of documents to "view" was made up of the top 50 ranked unseen documents, i.e. not including any that had been judged from any previous document set. An entry for each document consisted of:

- A header line.
  ⟨record_no⟩ ⟨docid⟩ ⟨weight⟩ ⟨document_length⟩ [⟨passage_length⟩] ⟨passage_length⟩
  was shown only for documents longer than 10K characters.

- A system generated title.
  Since, except for ZF and PT documents, records

had no distinct title field, a "title" for display in the hitlist window was made up from approximately the first 150 characters from the headline field (if any) and the text field.

- Query term occurrence information

A.2.4 "Seeing" documents

Documents were selected for "seeing" by double-clicking on the appropriate header line in the hitlist window. Each document was displayed in a pop-up, scrollable text window. The best passage, or the whole document if the same, was highlighted in cyan; query terms were highlighted in green. The document was displayed either at the start line of the best passage, or the line containing the first query term if there was no best passage.

In the case of documents longer than 10K characters, only the best passage was displayed with two or three lines of context around the start and end of the passage.

At the bottom of the text window were three buttons—"YES", "PASSAGE" and "NO" —to allow users to make a relevance judgment.

- "YES"

  Relevant: terms were extracted from the text field of the entire document.

- "PASSAGE"

  Relevant: terms were extracted only from the best passage.

- "NO"

  Not relevant.

Searchers had to make a relevance judgment before they could go onto to any other part of the interface.

A.2.5 Relevance judgments pool

The ranked hitlist information for all documents currently judged as relevant. Any member of the current relevance judgments pool that existed in a document set generated by a new search, had its weight set to that which it had in the latest document set; the display order was adjusted accordingly.

A.2.6 Expanding a query

Once the searcher had made one or more positive relevance judgments—"YES" or "PASSAGE" —the EXPAND button was selectable. Clicking on EXPAND caused the current query to be merged with all terms extracted from new relevant documents with relevance information adjusted accordingly and new weights and *RSV*s calculated. These terms were displayed in the query listbox in descending order of Robertson Selection value with a line drawn under the 20th term.

A.2.7 Removing terms

Terms were removed from the current query by double clicking on its entry in the query listbox. Although any term could be removed from the current query, its main use was for the removal of noise generated during the extraction of terms from relevant documents.

A.2.8 Quitting

Quitting was a two stage process performed by clicking the "END SEARCH" button twice.

- END SEARCH (1).

  This was taken as the end of the allowable search time, i.e. the end of the primary task. The top 20 terms from the current query iteration were displayed in the query listbox. Searchers were then allowed to modify this query by removing terms and / or adding new terms in the query entry box.

- END SEARCH (2).

  This marked the start of the secondary task, i.e. the generation of the ranked list of 1000 documents.

B Experimental Conditions

B.1 Searcher characteristics

Five (female) searchers were randomly assigned five searches each which were undertaken over a two day period. Three were research staff on the Okapi team and two were research students. Their ages ranged from late 20s to early 50s. All had extensive experience with the system but the interface was designed specifically for the interactive track and three of the searchers contributed to its specification.

None of the searchers had any specialist knowledge of any of the subject domains covered by the topics, they played the role of an intermediary searching on behalf of a remote end-user.

B.2 Task description/training

All but one of the searchers had participated in the TREC–3 interactive routing task, and all were aware of the different nature of the task definition for TREC–4. The searchers were given the opportunity to familiarize themselves with the interface by carrying out three or four trial searchers on the 25 non-interactive TREC–4 topics, over a period of a week prior to the actual test sessions.

Before conducting the test searches, the participants were issued with a set of guidelines which included information such as: the task definition as defined for the interactive track, procedures to follow in the event of a system crash, as well as suggestions on strategies which could be adopted for carrying out searches, e.g. to remove terms deemed as irrelevant from extracted lists of terms before carrying out an expanded search.

Searchers were also asked to fill in a search evaluation questionnaire for each search before proceeding on to the next search.

C Search process

C.1 Clock time

Times are given to the nearest tenth of a minute.

Mean	Median	Variance	Range[6]
30.8	30.6	12.3	22.0–41.7
30.3	30.6	7.5	22.0–34.3

C.2 Number of documents viewed (hitlist) and seen (full text)

C.2.1 Number of documents viewed

"Viewing" a document consisted of seeing a header line, a system generated title, and query term occurrence information as described in Appendix A.2.3.

The figures represent the percentage distance scrolled through the hitlist by the searcher.

Mean	Median	Variance	Range
54.18	20	1173.34	2–100

C.2.2 Number of documents seen.

"Seeing" a document consisted of showing the full record in a scrollable window.

Mean	Median	Variance	Range
31.12	30	97.2	13–52

C.3 Number of iterations

An iteration, i.e. a new query formulation, was taken to be marked by each 'search' command.

Mean	Median	Variance	Range
3.56	3	2.01	1–7

No. expands	Queries
0	3
1	20
2	2

[6] The maximum value of 41.7 was almost completely due to having to wait for around 30 minutes to view an FR record. The second row of figures have been calculated excluding this extreme value.

C.4 Number of terms used in queries

In all queries N = no adjacency and A = Adjacency.

C.4.1 Initial interactive

Type	Mean	Median	Variance	Range
N	2.32	1	4.98	0–7
A	2.40	2	3.83	0–8
All	4.72	4	8.96	1–12

C.4.2 Final interactive (primary task)

Type	Mean	Median	Variance	Range
N	12.84	13	33.72	1–20
A	3.80	3	7.08	0–10
All	16.64	19	25.74	3–20

C.4.3 Offline query (secondary task)

Type	Mean	Median	Variance	Range
N	12.80	13	33.83	0–20
A	4.24	3	8.02	0–10
All	17.04	19	26.87	3–20

C.4.4 "Phrases" defined by searchers

Phrases generated:	133
Phrases used:	121

C.5 Use of system features

+ terms defined with an adjacency operator

N terms defined with no adjacency operator

A all terms defined

Command	Mean	Median	Variance	Range
define - +	6.04	7	15.46	0–15
N	3.88	3	10.44	0–12
A	9.92	10	19.74	3–22
search	3.56	3	2.01	1–7
show	31.12	30	97.19	13–52
expand	0.96	1	0.21	0–2
remove	39.32	26	1146.14	0–113

C.6 Number of user errors

Data on user errors were not collected. However, there were three searches during which the system "crashed". These were undertaken by another searcher.

C.7 Search narrative for topic 236

The initial query consisted of the words 'sea', 'laws', 'disagree' and the adjacency term 'naval laws'. None of the documents resulting from this search were examined by the user since she could see from the hitlist that they were not relevant to the query. The user says of her lack of success: "In retrospect this was because I didn't know what I was looking for ...I could interpret the question—that it wanted information on disagreements on maritime law—but I wasn't sure what the disagreements might be—about age of ships, uniforms etc."

Two query terms were removed: 'disagree' and 'sea' and the user searched once again.

This time the user examined the first document and rejected it as not being relevant to the query. She then removed the term 'law' and searched again.

From the second document set five documents were looked at and one selected as relevant. Following this, the term 'naval laws' was removed from the query, leaving the query as the phrase 'laws of the sea'. The user searched again and selected two out of the five document seen as relevant. Following this she expanded her search. She gave as a reason for doing this: "I had found several relevant documents and had no idea what new terms to add myself so I thought it would be useful."

Of the expanded terms, the user deleted 46 terms. These included mainly proper names—surnames and the names of countries: 'Leslye', 'Arsht', 'Mariana', 'Samoa' and so on. Another search was then done and of 24 documents seen, six whole documents were selected as relevant, Seven relevant passages were selected and 11 documents were rejected. The user then expanded her search again.

Of the terms brought up by the second expansion, 60 were rejected. As with the terms rejected following the first expansion, these were mainly proper names, including: 'Gerasimov', 'Sevastopol', 'Kamchatka', 'McNaught'. The results of this search show the largest number of rejected terms (113) of any search session in the round. This may be due to the nature of the topic: most documents found concerned individual incidents of disagreement between two countries about sea territory and for this reason there were many geographical names in each document.

The user then ended the session and altered the final term-set first by deleting the term 'claims' and then by adding the term 'violated'. This left the following twenty terms as the final term-set:

Waters, laws of the sea, territory, miles, coast, seas, passage, warships, admiralty, seamounts, vessels, shipwreck, unassigned, eastward, Navy, offshore, maneuvred, ships, maritime, violated

The search took just under 26 minutes. Altogether, 35 documents were examined, of which nine whole documents and seven passages were judged as relevant. Most relevant documents referred to incidents where one country had violated the laws of the sea by crossing into another country's territory.

This search was one which started off badly with the user not sure what to look for but which picked up when the user expanded with just a couple of relevant documents and could see the type of information that the query was referring to.

Following is a breakdown of command usage on this search.

- Define(N): single term(s)
- Define(A): a "phrase"

Define(N)	Define(A)	Search	Show	Docset	Remove	Expand
3	2	6	35	6	113	2

D Search Evaluation: questionnaire results

1. How difficult was it to interpret the topic?

Easy	13	52%
Moderately easy	9	36%
Difficult	3	12%
Very difficult	0	0%
Total	25	100%

2. How difficult was it to generate initial search terms?

Easy	7	28%
Moderately easy	11	44%
Difficult	6	24%
Very difficult	1	4%
Total	25	100%

3. How difficult was it to find relevant items from the initial hitlist?

Easy	6	24%
Moderately easy	7	28%
Difficult	7	28%
Very difficult	5	20%
Total	25	100%

4. If you added new search terms in the course of the search, explain why?

Did not add terms	7	28%
Added terms to improve recall	14	56%
Added terms to improve precision	4	16%
Total	25	100%

5. If you chose to expand the search what led you to do so?

Did not expand	3	12%
Expanded to find more of the same type of item	11	44%
Expanded to find more precise items	5	20%
Expanded out of desperation	6	24%
Total	25	100%

6. If you removed terms from the extracted term lists on what basis did you do so?

Proper names	16	47%
Numbers	8	24%
Difficult	8	24%
Very difficult	2	5%

7. Was it difficult to determine what would constitute the offline query?

Easy	13	52%
Moderately easy	8	32%
Difficult	2	8%
Very difficult	2	8%
Total	25	100%

8. How would you rate the overall difficulty of the topic question?

Easy	8	32%
Moderately easy	7	28%
Difficult	7	28%
Very difficult	3	12%
Total	25	100%

9. How would you rate the success of your search?

Successful	9	36%
Moderately successful	9	36%
Not successful	7	28%
Total	25	100%

Table 8: Primary task evaluation results

Topic	Relevant	Retrieved	Rel_ret	Time	Precision	Recall	Rel/min
202	283	36	24	30	0.6667	0.0848	0.8000
203	33	8	4	30	0.5000	0.1212	0.1333
204	397	21	17	34	0.8095	0.0428	0.5000
205	310	19	9	29	0.4737	0.0290	0.3103
206	47	4	2	32	0.5000	0.0426	0.0625
207	74	14	11	31	0.7857	0.1486	0.3548
208	54	3	1	33	0.3333	0.0185	0.0303
209	87	7	1	32	0.1429	0.0115	0.0313
210	57	36	31	30	0.8611	0.5439	1.0333
211	323	19	19	31	1.0000	0.0588	0.6129
212	153	24	24	30	1.0000	0.1569	0.8000
213	21	20	10	31	0.5000	0.4762	0.3226
214	5	3	3	28	1.0000	0.6000	0.1071
215	183	13	10	31	0.7692	0.0546	0.3226
216	36	28	16	31	0.5714	0.4444	0.5161
220	24	1	1	21	1.0000	0.0417	0.0476
223	363	15	10	29	0.6667	0.0275	0.3448
227	347	42	36	30	0.8571	0.1037	1.2000
232	9	2	0	27	0.0000	0.0000	0.0000
236	43	16	11	24	0.6875	0.2558	0.4583
238	270	12	9	41	0.7500	0.0333	0.2195
239	123	16	13	30	0.8125	0.1057	0.4333
242	38	9	9	30	1.0000	0.2368	0.3000
243	69	41	4	30	0.0976	0.0580	0.1333
250	86	16	11	33	0.6875	0.1279	0.3333
All	137.40	17.00	11.44	30.3	0.6589	0.1530	0.3763
Micro average					0.6729	0.0833	

Table 9: Time boundaries

Measures calculated at 5 mins, 10 mins, 20 mins and at the end of the search.
Average end time = 31 mins.

D: time in minutes

J: number of iterations

E: number of official relevance judgments

F: number of searcher relevance judgments

G: number of official rels chosen by the searcher

H: precision (G / F)

I: recall (G / E)

K: number of full records seen by the searcher

L: number of official rels seen by the searcher as full records

M: number of official rels seen and rejected by the searcher

N: proportion of seen official rels that were rejected (M / L)

O: proportion of seen documents chosen by the searcher (F / K)

D	J	E	F	G	H	I	K	L	M	N	O
5	1.32	137.4	1.04	0.60	0.2600	0.0099	2.72	1.04	0.44	0.2500	0.1700
10	1.68	137.4	3.32	2.32	0.6260	0.0510	8.12	3.72	1.40	0.2742	0.2693
20	2.60	137.4	9.88	7.32	0.6807	0.1082	20.12	9.48	2.16	0.2400	0.3482
End	3.56	137.4	17.00	11.44	0.6589	0.1530	31.12	14.52	3.08	0.2447	0.3518
Micro average					0.6729	0.0833				0.2121	0.5463

Table 10: Iteration boundaries

Measures calculated at each iteration. The maximum number of iterations during any search was 7. For searches that had n iterations, n < 7, the values at iteration n were used for iterations n + 1 to 7.

J	D	E	F	G	H	I	K	L	M	N	O
1	2.64	137.4	0.00	0.00	0.0000	0.0000	0.00	0.00	0.00	0.0000	0.0000
2	14.32	137.4	5.88	4.20	0.5083	0.0355	11.44	5.52	1.32	0.2219	0.2786
3	20.92	137.4	10.16	6.56	0.6022	0.1022	18.88	8.72	2.16	0.2339	0.3065
4	25.24	137.4	13.04	8.84	0.6483	0.1198	23.80	11.44	2.60	0.2858	0.3377
5	28.20	137.4	15.76	10.72	0.6461	0.1395	27.96	13.72	3.00	0.2799	0.3400
6	29.44	137.4	16.56	11.08	0.6537	0.1446	29.56	14.12	3.04	0.2518	0.3465
7	30.24	137.4	17.00	11.44	0.6589	0.1530	30.80	14.52	3.08	0.2447	0.3518
Micro average					0.6729	0.0833				0.2121	0.5519

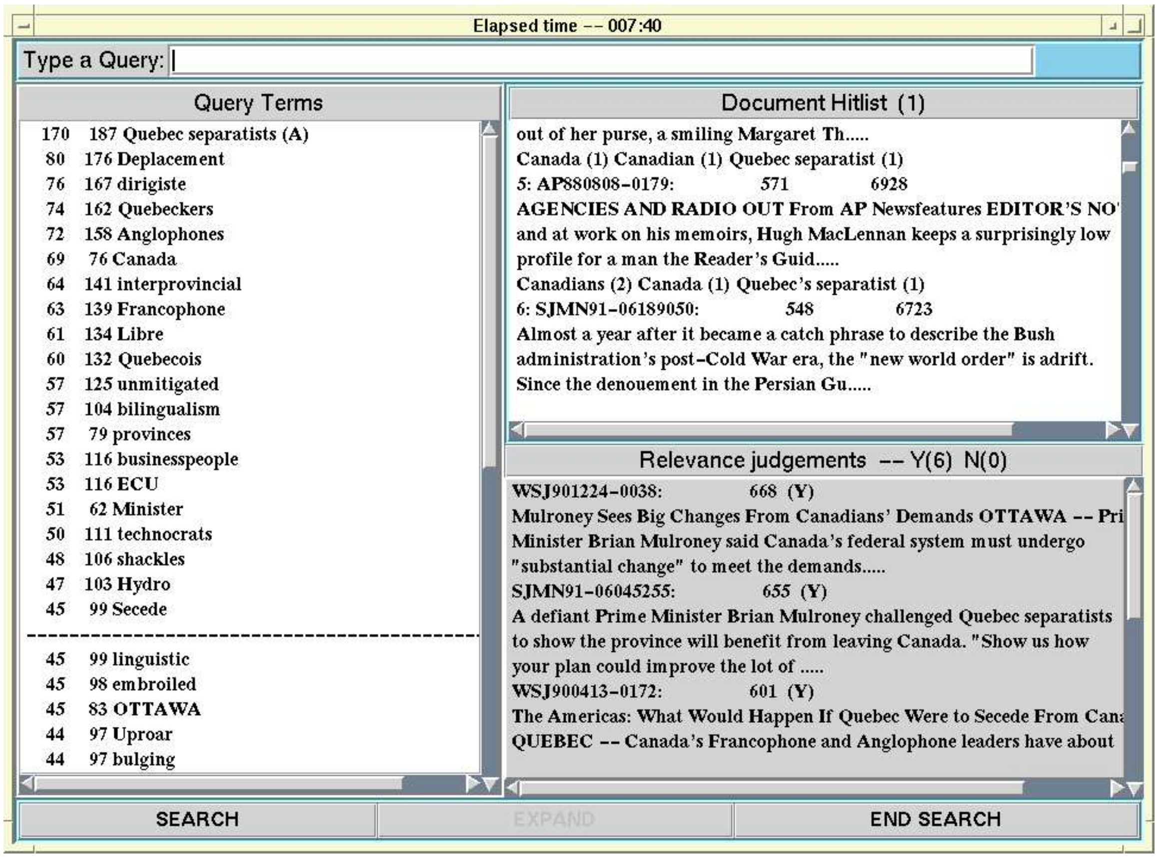


Figure 2: Interactive interface: main screen

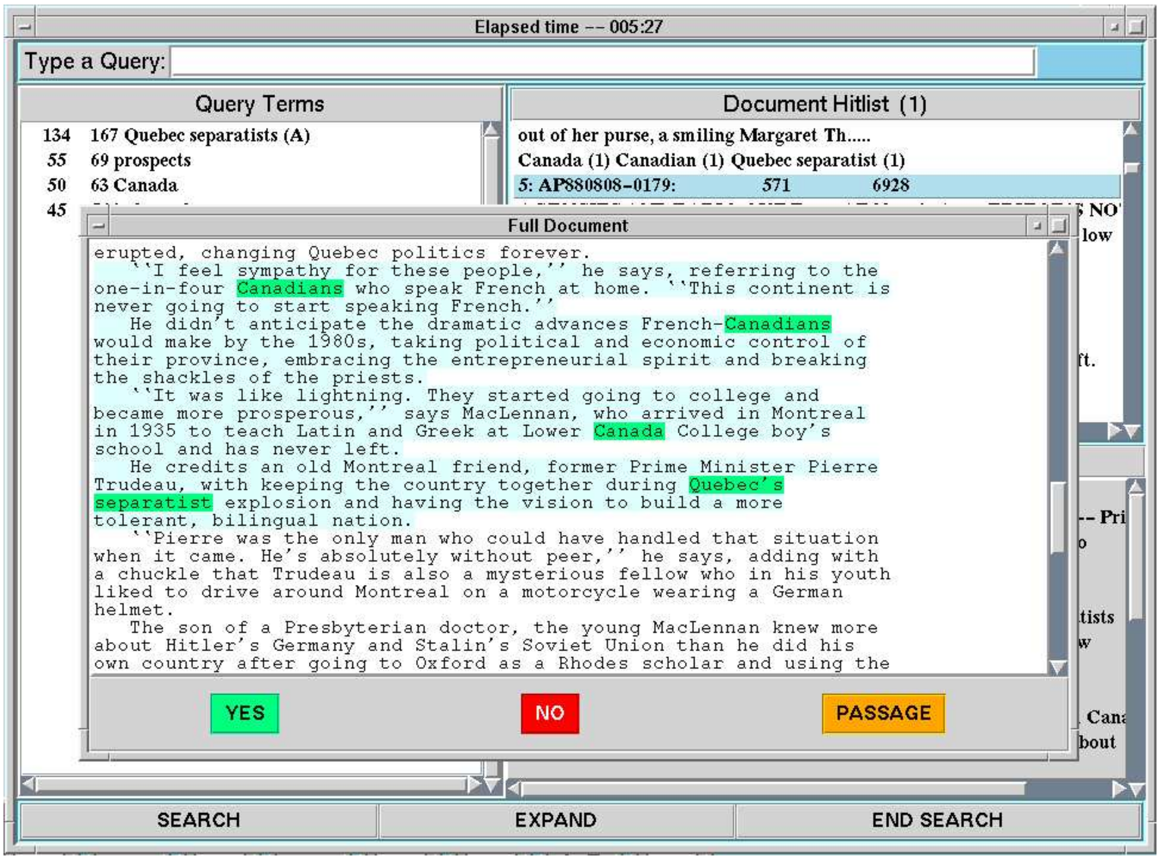


Figure 3: Interactive interface: full record display

E Guidelines for Searchers

Okapi TREC Interactive Searching Guidelines

Each searcher will be assigned five searches to be undertaken in the given order with a sufficient break in between each. The search topic must not be viewed before the search session itself. The time allowed is around 30 minutes for the primary task. The lapsed time during a session is displayed on the screen.

In the event of a system crash when a search is in progress, the search must be passed on to another searcher.

The primary task or objective is to find as many relevant documents during the search session for the given topic without too much rubbish and using whatever strategy is available in Okapi and which you consider to be appropriate.

Some suggestions on how to proceed follow.

- Since the topics consist of short questions it is suggested to generate as many terms as possible for the initial search.
- It may be useful to indicate phrases wherever possible by typing a '+' after consecutive terms.
- Once a hitlist has been generated and items viewed, it may be fruitful to add further terms or phrases at this stage before any query expansion is undertaken. Trying to add your own new terms to a list of extracted terms once query expansion has been chosen, may not be as productive since their inclusion will depend on their relative weights.
- It is usually more effective to expand a search only after several items have been deemed relevant. If you want to expand a search, you must allow enough time, e.g. before the final ten minutes.
- The prime purpose of the relevance judgment is to indicate the relevance of an item to the topic or question. Making relevance judgments for the purpose of query expansion alone is not encouraged.
- Remove any terms deemed irrelevant from extracted lists of terms before carrying out an expanded search. The top twenty terms will be used to generate a new hit list.
- To determine the final query for the secondary task, it may be appropriate to select the final list of terms, your original query only or a combination of both including any new terms you may want to add at that point.

Please fill in a search evaluation on completion of a search before proceeding to the next one.