PERGAMON

Information Processing and Management 36 (2000) 95–108

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

Experimentation as a way of life: Okapi at TREC

S.E. Robertson*, S. Walker[1], M. Beaulieu[2]

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

Abstract

The Okapi system has been used in a series of experiments on the TREC collections, investigating probabilistic models, relevance feedback and query expansion, and interaction issues. The TREC-6 ad hoc task was used to test an application of a new relevance weighting formula, which takes account of documents judged nonrelevant. The application was to a form of blind feedback (using the top-ranked documents from an initial search to improve the query formulation for a subsequent search, without actual relevance feedback, on the assumption that these top-ranked documents are likely to be relevant). In the routing task, the problem is one of query optimization based on a training set with known relevant documents; investigations for TREC-6 included using a form of simulated annealing for this purpose. A significant feature of this work is the need to avoid overfitting of the training sample. In the interactive track, methodology remains the major problem: we do not yet know how to conduct controlled laboratory experiments which provide good information about information retrieval interaction. The Okapi team has been particularly interested in the relation between the functionalities associated with relevance feedback and the ability of searchers to make use of these functionalities. TREC provides an excellent environment and set of tools for investigating automatic systems; its value for interactive systems is not yet proven. © 1999 Elsevier Science Ltd. All rights reserved.

* Corresponding author. Present address: Microsoft Resarch Ltd, St George House, 1 Guildhall Street, Cambridge CB2 3NH, UK. Tel.: +44-1223-744-769; fax: +44-1223-744-777.

E-mail address: ser@microsoft.com (S.E. Robertson).

[1] Present address: Microsoft Research Ltd, St George House, 1 Guildhall Street, Cambridge CB2 3NH, UK.

[2] Present address: University of Sheffield, UK.

PII: S0306-4573(99)00046-1

1. Introduction

1.1. The Okapi projects

Okapi is the experimental text retrieval system which has been under continuous experimentation at City University London for the last 10 years. The main concerns of the last few years have been weighting and ranking methods and relevance feedback, and user interaction, with particular reference to user perception and use of advanced functionality, such as query expansion. An extensive account of this work was recently published in the form of a special issue of *Journal of Documentation* (Robertson, 1997).

1.2. Okapi at TREC

The TREC programme as a whole is described and discussed elsewhere in this issue.

The Okapi team at City University has taken part in every round of TREC. The opportunity afforded by this programme to conduct a wide variety of experiments has been the major formative influence on the work of the team over that period, and has encouraged and made possible substantial developments both in system design and in underlying models.

At the same time, the team is aware of the limitations of the kind of methodology embodied in TREC. This awareness stems partly from earlier and parallel experiments in operational environments, with real users with real information needs.

This paper is concerned with both the successes and the limitations of this series of experiments. It therefore attempts to address, in the context of the City work, the question: What are the parameters which define the value of experiments such as TREC?

The work with Okapi in various rounds of TREC has been described in the official proceedings (Beaulieu et al., 1997, and earlier; Beaulieu & Gatford, 1998; Walker, Robertson & Boughanem, 1998). A paper also appeared in an earlier special issue of Information Processing and Management (Robertson, Walker & Hancock-Beaulieu, 1995), and another in the issue of Journal of Documentation mentioned. Some work in the interactive track of TREC was discussed in a JASIS paper (Beaulieu, Robertson & Rasmussen, 1996).

2. Probabilistic models and weighting functions

Okapi is based on a probabilistic model of retrieval, essentially a development of the Robertson/Sparck Jones model (Robertson & Sparck Jones, 1976). Participation in TREC has led to extensive development of the basic model and a number of variant weighting functions.

The basic Robertson/Sparck Jones weight for a term is:

$$w^{(1)} = \log\frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5),} \tag{1}$$

where 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. This is a component of the BM25 function defined below. In the absence of relevance information ($R = r = 0$) it defaults to a collection-frequency weight (IDF); if the user judges some documents to be relevant this information can be fed into the formula. It may also make use of 'blind' or 'pseudorelevance' feedback, where no real relevance information is available, but an initial search is conducted and the top few documents are *assumed* to be relevant.

Part of the TREC-6 work discussed below involved replacing Eq. (1) with a more general function which takes account of documents judged nonrelevant as well as those judged relevant (see below).

2.1. Term selection

Much recent work involves query expansion, which involves using relevance (or pseudorelevance) information to identify new terms which may be added to the query. In general, we make use of a term selection value (TSV) for this purpose (Robertson, 1990), which is not the same as the weight. Candidate expansion terms (essentially all terms extracted from relevant documents) are generally ranked in TSV order, and a certain number of terms are taken from the top of this ranking.

The usual formula for TSV is

$$\mathrm{TSV} = r \times w^{(1)} \tag{2}$$

2.2. Term frequency and document length

The most substantial change in the model was the incorporation of within-document term frequency, within-query term frequency and document length. This work was done over TREC-2 and TREC-3, and resulted in considerable success at TREC-3 (Robertson et al., 1995).

The resulting weighting functions can be represented as follows. This is a somewhat simplified version, but it incorporates all the variations which have been found useful in the *tf*, *dl* and *qtf* components.

Then the (slightly simplified) Okapi BM25 document weighting function is

$$\sum_{T \in \mathcal{Q}} w^{(1)} \frac{(k_1 + 1)tf}{K + tf} qtf, \tag{3}$$

where $\mathcal{Q}$ is a query, containing terms T, K is $k_1((1-b)+b \times dl/avdl)$, k_1 and b are tuning parameters for which suitable values have to be discovered by experiment. For the TREC-6 experiments, typical values of k_1 and b were 1.0–2.0 and 0.35–0.75 respectively, *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 $\mathcal{Q}$ was derived, *dl* and *avdl* are the document length and average document length (arbitrary units), respectively.

Apart from the changes to $w^{(1)}$ discussed below, this BM25 formula has remained more-or-less fixed since TREC-3.

2.3. RGS formula

The modified version of $w^{(1)}$ used in TREC-6 was first introduced by Robertson and Walker (1997). The term weight is divided into two parts, one of which depends on the probability of the term occurring in a relevant document and the other on the probability of the term occurring in a nonrelevant document. Then each is again divided, into a prior estimate and an evidence-based estimate (depending on items judged for relevance by the user). The result is a somewhat complex formula, as follows.

$$w^{(1)} = \frac{k_5}{k_5+\sqrt{R}}\left(k_4+\log\frac{N}{N-n}\right)+\frac{\sqrt{R}}{k_5+\sqrt{R}}\log\frac{r+0.5}{R-r+0.5}-\frac{k_6}{k_6+\sqrt{S}}\log\frac{n}{N-n}$$
$$-\frac{\sqrt{S}}{k_6+\sqrt{S}}\log\frac{s+0.5}{S-s+0.5} \tag{4}$$

Here we have two further pieces of data, S is the number of documents known to be nonrelevant to a specific topic; s is the number of nonrelevant documents containing the term; and three further tuning parameters, k_4–k_6. k_4 is a starting point for the prior relevance probability (or rather, log-odds); k_5 determines the relative weight of prior and evidence in the case of the relevance probability, and k_6 does the same for nonrelevance.

There are several possible applications for such a formula (Robertson & Walker, 1997), but the main use investigated in TREC-6 was a variant on the blind feedback idea. Here the principle was to do an initial retrieval, take the top few (~10) as relevant, leave a gap (~500), and then take a further block (~500) as nonrelevant — hence the nomenclature RGS formula.

In line with this modification of $w^{(1)}$, we may propose a modification[3] to the TSV formula (2)

$$\text{TSV} = (r/R - \alpha s/S) \times w^{(1)}, \tag{5}$$

where $\alpha \in [0,1]$ and r, R, s and S are as above.

2.4. Experimental results

The specific experimental results reported below were obtained during the course of TREC-6. Some are based on TREC-6 data, and some on earlier data, as indicated in the tables.

3. Automatic ad hoc experiments

3.1. Collection frequency

One of the reasons for introducing Eq. (4) was an anomaly which arises in some

[3] Due to M. Boughanem.

circumstances in Eq. (1). In the absence of relevance information, the latter reverts to a collection-frequency weight, as indicated. However, if the term in question is a very frequent one (occurring in more than half the collection), (1) actually gives a negative weight, which seems counterintuitive. (4) does not do so if the starting constant k_4 is zero or positive. (If k_4 is zero, (4) is otherwise very similar to (1).)

One of the first experiments in TREC-6 was therefore to try (4) without relevance information, with different values of k_4. Unfortunately from the point of view of the initial argument, the best results in these circumstances were obtained with a *negative* k_4, which reintroduces the negative collection frequency weight in certain unusual conditions. Table 1 gives some results showing this effect. The result does, however, confirm that a closer look at the prior estimate of the relevance weight (which is in effect what collection frequency weight is) is in order.

(In this and subsequent tables, AveP refers to noninterpolated average precision over the top 1000 ranked documents. P30 refers to precision at 30 documents retrieved, and Rcl refers to recall at 1000 documents. The particular set of topics and documents used for the test are specified in each table.)

3.2. Blind reweighting

The RGS formula (4) may be applied with pseudorelevance information to reweight the original query terms. We have some evidence from TREC-6 that this may be an effective procedure — see Table 2.

3.3. Blind expansion

Given that the document scoring methods used since TREC-3 have been very successful, quite possibly approaching the limit of what can be achieved by matching only on the original query terms, much of the effort of the last few years has been on the use of pseudorelevance feedback to expand the query. In recent TRECs this has been an area explored by many participants, with varying degrees of success. We have done this using the old formula in the past; for TREC-6 we used the RGS formula.

Our experience suggests the following:

1. Blind expansion can be a useful device, but is somewhat erratic.

Table 1
Effect of k_4.
43 TREC-5 topics with at least 10 rel docs; TREC-5 documents (disks 2 and 4) (L = long topics, S = short topics)

Method	AveP	gain %	P30	Rcl
L: $k_4 = 0$	0.219	0.0	0.363	0.470
L: $k_4 = -0.7$	0.225	2.7	0.364	0.482
S: $k_4 = 0$	0.150	0.0	0.281	0.372
S: $k_4 = -1$	0.159	6.0	0.268	0.389

Table 2
Effect of blind reweighting.
50 TREC-5 topics; TREC-5 documents (disks 2 and 4) (L = long topics, S = short topics)

Method	AveP	gain %	P30	Rcl
L: no reweighting: $R = G = S = k_4 = 0$	0.230	0.0	0.389	0.471
L: $R = 10$, $G = 500$, $S = 1000$, $k_4 = 0$, $k_5 = 2$, $k_6 = 64$	0.242	5.2	0.335	0.519
L: as above except $k_4 = -1$	0.247	7.4	0.331	0.519
S: No reweighting: $R = G = S = k_4 = 0$	0.164	0.0	0.247	0.369
S: $R = 4$, $G = 500$, $S = 500$, $k_4 = 0$, $k_5 = 2$, $k_6 = 128$	0.178	8.5	0.256	0.403
S: as above except $k_4 = -1$	0.184	12.2	0.261	0.409

2. In particular, it works well for some queries but disastrously for others; its average performance over a set of topics depends very strongly on the make-up of the set.
3. On the whole, it is better to do the blind expansion from a large database. That is, whatever the target database for retrieval, it is best to do the initial search on as large a database as possible; there is then more chance that the top ten (or whatever) documents contain a high proportion of relevant items. The expanded query can then be applied to the target database for the final search.
4. As one might expect, there is more scope for expanding short initial queries than long ones (however, the variability may be greater).

Some sample results will indicate the kinds of effects that might be observed. It must, however, be emphasized that we do not regard these results as stable, or the general principle of blind expansion as proven.

The first set of results (Table 3) are from TREC-6, and use the RGS formula. Expansion was based on a search on a large database, consisting of the cumulated data from TREC disks 1–5.

The second set (Table 4) are from an older collection, and use the old formula. However, they provide some comparison between blind expansion and expansion using real relevance information. They are based on dividing the collection of documents in half (‘even’ and ‘odd’ halves), obtaining any feedback information from the odd half, and running the test on the even half. Blind expansion uses all the top 10 documents retrieved on the odd half; ‘real relevants’ are those in the top 10 judged relevant. Expansion uses the top 24 terms.

Table 3
Effect of blind expansion−1.
50 TREC-6 ad hoc topics; TREC-6 ad hoc documents disks 4 and 5 ($k_1 = 1.2$, $b = 0.75$)[a]

Method	AveP	gain %	P30	Rcl
Blind expansion, $k_4 = 0$, $k_5 = 1$, $k_6 = 64$,	0.271	8.4	0.351	0.561
Unexpanded, $k_4 = -0.7$	0.250	0	0.337	0.531

[a] Terms used in expansion run obtained from: initial search on disks 15, with $k_3 = 7$ (k_3 is a tuning parameter in the more complex version of BM25 which modifies the effect of the qtf component). $R = 10$, $G = S = 500$, $\alpha = 0.15$; top 30 terms (but query terms loaded).

Table 4
Effect of blind expansion — 2.
TREC topics 51–200; Even half of disks 1 and 2 ($k_1 = 2.3,\ b = 0.7,\ k_3 = 20^{a}$)

Method	AveP	gain %	P30	Rcl
Real relevants, expansion and reweighting	0.336	24.9	0.517	0.713
Blind expansion and reweighting	0.318	18.2	0.494	0.714
Real relevants, reweighting only	0.282	4.8	0.456	0.659
Baseline	0.269	0	0.442	0.651

[a] k_3 is a tuning parameter in the more complex version of BM25 which modifies the effect of the qtf component.

3.4. Very large corpus

We took part in the VLC track of TREC-6, in order to establish whether Okapi could operate on that scale. The results are not particularly noteworthy, except for the observation (made by us and every other VLC participant) that the performance in terms of precision at a fixed document cutoff was better in the VLC itself than in the 10% sample collection.

This result is compatible with the observation above, that in general blind expansion is better done from a large collection. It is an interesting observation in its own right. We have put forward a specific explanation for the result, based on the distribution of scores (retrieval status values) over relevant and nonrelevant documents, and how this distribution might be expected to vary between the whole corpus and the sample. The hypothesis is discussed by Hawking, Thistlewaite and Harman (1998).

4. Routing and filtering: query optimization

The basic approach to successive TREC routing tasks has been to use the training information to devise an optimal search formulation. We may see this as the exploration of a very large space of possible solutions. Essentially any search term may be included in the formulation, at any weight (one might also consider other components, such as phrases or Boolean combinations). In principle, we could envisage trying out every possible combination, but the search space is far too large for such exhaustive exploration, so the strategy has been to

1. devise heuristics to allow the exploration of a good range of likely areas of the space;
2. increase the efficiency of exploration;
3. use increasingly powerful equipment.

These devices have enabled us to explore more of the space with each successive TREC.

However, this process also brings into focus a fundamental problem with the idea of an exhaustive search, which is becoming more and more evident as we succeed in expanding our heuristics. This is the problem of overfitting: if we succeed too well in matching the characteristics of the training set and learning precisely what would work well on that data, we have probably specialised too much. That is, we will have learnt to distinguish the relevant documents *in the training set* from the nonrelevant ones also in the training set, using whatever characteristics may be used for this purpose, even if these characteristics are accidental or in some other way peculiar to the training set. What is required for effective routing is to identify those characteristics which will be equally good on the next set of documents to arrive.

Thus we have found that in successive TRECs, we have had to work harder to reduce the danger and extent of overfitting.

4.1. Basic procedure

All experiments were conducted with the original formula (1) for $w^{(1)}$ and the original term selection value formula (2).

Terms are extracted from known relevant documents, and ranked according to TSV. A fixed number of candidate terms (the term pool) is taken from the top of this ranking and subject to selection and/or reweighting procedures. A large number of sets of terms and associated weights are formed from the pool and the performance of each set is measured on a particular database (the same as or different from the one used for initial extraction). One particular performance measure is used to score the set; in TREC-6, the only measure we used for this purpose was the noninterpolated average precision on the top 1000 documents (a variety of other measures have been tried in the past, but average precision appears to be generally effective). In principle, the best (highest scoring) set of terms and weights then becomes the definitive search formulation.

However, this is the point at which the problem of overfitting becomes apparent. Our procedure here is to repeat the exercise with a number of different methods of generating the new sets of query terms and weights. Each method produces its own candidate 'best' formulation. Rather than taking the best out of these, we now merge the resulting formulations in a manner given below. This procedure appears to provide a degree of insulation against the overfitting problem.

4.2. Space exploration

This essentially involves an iterative procedure for generating new search formulations by modifying old ones.

In earlier TRECs, we confined our exploration of the space to *selection* from the pool, keeping the weights as originally assigned by the formula, by simply adding pool terms to, or deleting them from, the search formulation. This procedure tended to result in relatively small formulations ($\tilde{1}$0–20 terms), in considerable contrast to the procedures used by some other members of the TREC community, which resulted in massive expansion. In TREC-5, we did some reweighting in addition to selection, which we found moderately beneficial and which had the effect of increasing our query sizes a little. All of these procedures were deterministic in the sense that every step involves a well-defined, fully specified change, and there are well-defined, fully specified rules for accepting or rejecting a change.

We might, by contrast, consider some form of stochastic procedure. There are several such procedures on offer: genetic algorithms for example. In TREC-6, we conducted some experiments with another stochastic procedure, namely simulated annealing. Essentially this involves random changes, and a randomised rule about accepting changes, which will sometimes accept a change that produces a reduction in the score. The object of this process is to encourage the candidate solution to escape from what might be a local maximum, and allow it to explore other possible solution-regions. The extent to which this is allowed to happen depends on the 'temperature' of the annealing process; this temperature is dropped in successive steps, until a 'quench' stage at zero temperature.

Clearly there are many different parameters to be set before such a procedure is completely specified. Our TREC-6 experiments merely scratched the surface, but were not in any case particularly encouraging — predictive results from the search formulations generated by simulated annealing varied rather wildly, and in the end we did most of our exploration by deterministic means, followed by a relatively mild form of annealing.

4.3. Merging runs

Merging search formulations involves merging the sets of terms, and weighting each term with the sum of its weights in the source formulations. If the source formulations have very different weight ranges (as happened often with simulated annealing), then the term weights from one formulation are first normalised by the median weight of terms in this formulation.

When the term pool has been derived from half the training set of documents and then optimized on the other half, the reverse procedure is also performed. This provides a pair of search formulations for merging. Further formulations for merging are generated by other changes in the procedures.

4.4. Automatic routing results

These are given in Table 5; city6r1 and city6r2 are the two official runs for TREC-6. These were made with a form of passage retrieval used in recent TRECs, not discussed further in this paper, which gives a small but consistent benefit; their performance without passage searching is also shown. The four columns beside AveP compare the performance on each topic with the median for that topic in the official TREC runs.

Both sets of submitted queries (city6r1 and city6r2) were derived using only the TREC-5 routing database and the TREC-6 filtering training relevance judgments. Had time allowed we should probably have used some additional pre-TREC-5 training information for some of the

Table 5
Automatic routing results.
TREC-6 routing topics; TREC-6 routing documents

Run	AveP	≥ med.	best	< med.	worst	P30	Rcl
city6r1	0.408	41	7	6	0	0.548	0.809
(no passages)	0.399	41	1	6	0	0.545	0.802
city6r2	0.378	39	5	8	0	0.523	0.760
(no passages)	0.368	37	3	10	0	0.515	0.753

topics with few known relevant FBIS documents. The difference between the two sets is that for city6r1 the terms came from one half of the database and the optimization was done on the other half; whereas for city6r2 the whole database was used both as term source and for optimization. Experiments with TREC-5 routing data had suggested that the former method was likely to give slightly better results, although the relatively small number of TREC-6 training judgments made it dangerous to assume that this would still hold. Hence it is quite surprising that the cityr1 result turned out so much the better of the two. However, there were other differences between them. Both sets of queries were formed by merging a number of query sets, but 24 (12 pairs) were used to form city6r1 and only six for city6r2; city6r1 used four deterministic weight variation passes with just a final 'quench' stage; city6r2 had three deterministic passes followed by four stages of simulated annealing.

All the optimization runs started with a term pool of size 100 (in previous TRECs we had found a small gain from using up to 300 terms, but the simulated annealing would have been too slow on sets of this size). City6r1 ended with a mean of 138 terms per query and city6r2 with 86, but many terms had very low weights and the queries could probably be reduced by 25 percent or more without greatly affecting results.

4.5. Filtering

We also submitted entries for the filtering track, based on our routing runs. These were determined by rerunning the routing formulations against a training database and fixing a threshold by determining the point in the ranking at which the required utility function was maximized. This procedure appeared to work extremely well; our filtering results (like our routing results) were among the best at TREC-6. However, this is not very helpful when we consider the next step to be taken (see below).

4.6. Discussion: adaptive filtering

We did not take part in the new adaptive filtering track in TREC-6, but hope to do so in TREC-7. It presents a number of very interesting challenges: essentially we have to start with a form of ad hoc retrieval (with no known relevance information); then as we acquire a little relevance information, we may apply one of the relevance weighting formulae; then with more relevance information, we may start an iterative optimization procedure. Throughout this process, we have to have a filtering threshold. The question of how to set thresholds at the earlier stages has not yet been tackled.

5. The ongoing challenge: interactive system evaluation

From the outset the Okapi team has been a major participant in the interactive rounds of TREC. The system was originally designed for carrying out operational testing with naive untrained users and its query expansion facility has provided a very rich environment for conducting interactive experiments. However, unlike the successes in the automatic ad hoc and routing experiments, progress in the interactive track has been very slow and the Okapi results to date, in common with those of other players, have been limited.

The heart of the problem is that an appropriate methodological approach for interactive system evaluation has yet to be defined. Whereas the main TREC experiment has closely followed the Cranfield retrieval test model, the experimental design for interactive searching has been in itself exploratory and experimental. At the outset some attempt was made to accommodate interactive evaluation within the TREC framework. However, after the first three rounds it became apparent that due to the differences in the experimental variables and conditions, the automatic and interactive experiments could not produce comparable results. Moreover, the fact that automatic methods for query construction could outperform queries formulated by a human searcher in a laboratory setting did not in itself offer any insight into designing more useable systems. Since any realistic retrieval task has to include human intervention at some level, the challenge remains how best to incorporate or simulate the human searcher in an evaluative laboratory experiment.

5.1. Design of the TREC-6 interactive track

Full details of the design are given in the proceedings (Voorhees & Harman, 1998). The following gives an outline only.

When a separate interactive track was introduced in TREC-4, the stated goal was to investigate searching as an interactive task by examining the process as well as the outcome. Efforts to establish a more appropriate framework have concentrated on two elements: firstly redefining the interactive search task itself, and secondly devising an experimental design matrix to minimise the effect of possible interactions between searchers, topic and system elements. For TREC-6, six topics were chosen which contained multiple aspects, in the expectation that relevant documents would tend to address only some of the aspects. Searchers were allocated 20 min per topic, to find relevant documents which covered as many different aspects as possible. The rationale was to set a more realistic and discriminating task which encouraged searchers to browse in order to make more specific relevance judgements on selected documents. The results were measured in terms of aspectual recall and precision; in the case of the former measure at least, this assessment thus relates to the set of documents comprising the search outcome, rather than being a simple accumulation based on individual documents.

With regard to the control of the variant components, a matrix for allocating the order of searchers, topics, and systems was devised. At each site, the local system was compared with NIST's ZPRISE system without relevance feedback as a control.

5.2. Incremental query expansion

Since the same bestmatch weighting function (BM25) has been used for all the interactive rounds, the focus of the Okapi TREC interactive experiments has been primarily on the user searching process rather than on the search engine per se. On the one hand the aim has been to determine how the system could support the user in formulating queries and in making relevance judgements; and on the other hand to assess how and when the user could best intervene in the search process. Based on the general assumption that a highly interactive interface environment would be the most beneficial, the interface design and overall experimental approach has been to increase the flexibility of the user interface and to provide more opportunities to interact with the system.

In previous interactive rounds users could manipulate the query by removing system extracted query terms or adding new terms. However the main difference introduced in TRECs 5 and 6 was that query expansion was presented to the user in an incremental fashion. In incremental query expansion, once a searcher makes a positive relevance judgement, extracted terms are automatically added to the working query and current query terms are dynamically reweighted and reordered, if the necessary conditions for term selection values are met. The results of this process, to a maximum of twenty terms, are displayed to the user immediately in the working query window. The intention was to make the relevance feedback process more visible by enabling the searcher to relate positive relevance judgements more closely to the expanded query. In TREC-6 some minor adjustments were made to the incremental query expansion and term selection conditions in order to minimise seemingly erratic changes to the working query and the need for the user to remove too many unwanted terms. For example TREC-6 took account of the term selection value of all the terms that had been in the working query at any time and not only those that had appeared in the most recent working query formulation.

5.3. Highlighting best passages

In addition to the features to facilitate the formulation of queries, attention was also paid to the presentation of search results and documents. The highlighting of best passages of documents was first introduced in TREC-4 to enhance the display and viewing of documents, to enable the searcher to make relevance judgements more quickly and also to choose between the full document and best passage only for relevance feedback. Passage retrieval is deemed particularly useful for long documents and in the case of documents dealing with several topics, it makes it possible to directly display the section most appropriate to the query.

5.4. Search performance and user searching behaviour

It was anticipated that the current precision orientated interactive task would be somewhat taxing for the Okapi system. Past experiments within and outside TREC have shown that query expansion is effective at presenting more of the same rather than finding items that are slightly different or related to those already retrieved. The comparative results in TREC-6 show that the ZPRISE system achieved better precision than Okapi (Okapi $R = 0.400$, $P = 0.706$; ZPRISE $R = 0.381$, $P = 0.809$). It would thus appear that searchers could generate more effective queries without the help of query expansion.

Although searchers found both systems easy to use, a greater proportion were satisfied with the search outcome in Okapi (50%) than in ZPRISE (37%). This discrepancy between users' perception and actual search performance could be indicative of users' satisfaction with the level of support provided by the system rather than the actual search outcome. Searchers on Okapi did seem to be more dependent on the system. Although the number of terms in the initial query (Okapi 3.16, ZPRISE 3.68) and the total number of user generated query terms (Okapi 6.00, ZPRISE 6.95) were not significantly different for the two systems, Okapi searchers introduced less than half the number of terms in the query after the first iteration than for ZPRISE (2.33, 5.54).

The adjustment to the incremental query expansion facility may have had some effect in that users generated and removed fewer terms in the course of the searches but on average they undertook the same number of iterations as in TREC-5, 3.38 and 3.63 respectively. But the final number of query terms was also substantially reduced from a mean of 18.21 in TREC-5 to 4.84 in TREC-6; the mean for ZPRISE was even less at 3.79.

With regard to the display of the best passages, Okapi searchers made more positive relevance judgements than on ZPRISE but these were made on the full record rather than on the best passage only.

It is premature to draw any firm conclusions on these results. Clearly in terms of system performance, diagnostic analysis is required to determine how to optimise incremental query expansion and how to use best passage retrieval for relevance feedback in the context of the interactive search task as it is currently defined. As an initial step it is our intention to undertake more direct comparisons in the use and non use of relevance feedback in the next round of TREC.

6. Conclusions

TREC provides a marvelous environment and set of tools for conducting highly informative and productive experiments concerning various automatic tasks in the general context of information retrieval. Without any question, the state of the art of text retrieval has advanced substantially in the past six years of TREC, both through the participants' direct efforts towards TREC itself and through theirs and others' use of TREC materials outside the immediate TREC environment. It is also clear to us that there is much more work to be done. Even if TREC were to cease as a collective endeavour after TREC-7 or 8, the accumulated material would continue to serve as the basis for much useful work for many years to come. However, it is also apparent that both the stimulus and the discipline of the full TREC experimental environment have immense value, and we hope that TREC itself will continue for a while yet.

By contrast with the work on automatic methods, TREC has not had the same impact for the evaluation of interactive systems. Whilst we believe that the effort to establish an appropriate framework for the interaction area is essential and most worthwhile, it would appear that there is little chance that this could become a major activity within TREC. The time may have come to consider a new type of collective endeavour or forum which could incorporate many other forms of experiments, for example with real users in live-use environments, as well as laboratory experiments with very different tasks and measures of outcome.

References

Beaulieu, M. M., & Gatford, M. J. (1998). Interactive Okapi at TREC-6. In E. M. Voorhees, & D. K. Harman, *Proceedings of the sixth Text REtrieval Conference (TREC-6)* (pp. 143–167) NIST Special Publication 500-240.

Beaulieu, M. M., Robertson, S. E., & Rasmussen, E. (1996). Evaluating interactive systems in TREC. *Journal of the American Society for Information Science*, *47*, 85–94.

Beaulieu, M., Gatford, M., Huang, X., Robertson, S., Walker, S., & Williams, P. (1997). Okapi at TREC-5. In E. M. Voorhees, & D. K. Harman, *Proceedings of the fifth Text REtrieval Conference (TREC-5)* (pp. 143–166). Gaithersburg, MD: NIST NIST Special Publication 500-238.

Hawking D., Thistlewaite P. & Harman D. (1999) Scaling up the TREC collection. *Information Retrieval, 1*, 113–137.

Robertson, S. E., Walker, S., & Hancock-Beaulieu, M. M. (1995). Large test collection experiments on an operational, interactive system: Okapi at TREC. *Information Processing and Management*, *35*, 345–360.

Robertson, S. E. et al. (1995). Okapi at TREC-3. In D. K. Harman, *Overview of the Third Text REtrieval Conference (TREC-3)* (pp. 109–126). Gaithersburg, MD: NIST NIST Special Publication 500-225.

Robertson, S. E., & Walker, S. (1997). On relevance weights with little relevance information. In N. J. Belkin, A. D. Narasimhalu, & P. Willett, *SIGIR 97. Proceedings of the 20th International Conference on Research and Development in Information Retrieval, Philadelphia* (pp. 16–24). New York: ACM.

Robertson, S. E., & Sparck Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science*, *27*, 129–146.

Robertson, S. E. (1990). On term selection for query expansion. *Journal of Documentation*, *46*, 359–364.

Robertson, S. E. et al. (1997). *Journal of Documentation, 53* (1) 10 papers, pp. 3–87.

Voorhees, E.M. & Harman, D.K. (1998). *Proceedings of the sixth Text REtrieval Conference (TREC-6)*. NIST Special Publication 500-240.

Walker, S., Robertson, S. E., & Boughanem, M. (1998). Okapi at TREC-6: automatic ad hoc, VLC, routing and filtering. In *Proceedings of the sixth Text REtrieval Conference (TREC-6)* (pp. 125–136). Gaithersburg, MD: NIST Special Publication 500-240.