ANU/ACSys TREC-6 Experiments

David Hawking, Paul Thistlewaite and Nick Craswell
Co-operative Research Centre For Advanced Computational Systems
Department Of Computer Science
Australian National University
{dave,pbt,nick}@cs.anu.edu.au [*]

February 19, 2007

Abstract

A number of experiments conducted within the framework of the TREC-6 conference and using a completely re-engineered version of the PArallel Document Retrieval Engine (PADRE97) are reported. Passage-based pseudo relevance feedback combined with a variant of City University's Okapi BM25 scoring function achieved best average precision, best recall and best precision@20 in the Long-topic Automatic Adhoc category. The same basic method was used as the basis for successful submissions in the Manual Adhoc, Filtering and VLC tasks. A new BM25-based method of scoring concept intersections was shown to produce a small but significant gain in precision on the Manual Adhoc task while the relevance feedback scheme produced a significant improvement in recall for all of the Adhoc query sets to which it was applied.

1 Introduction

The work reported here comprises a number of text retrieval experiments conducted within the framework of TREC-6 and addressing questions of interest in the following research areas: Scalable information retrieval; Relevance Feedback; Distance-based relevance scoring; Selective Dissemination of Information and Automatic Query Generation. ANU/ACSys completed Automatic and Manual Adhoc, Filtering and VLC tasks.

1.1 Relevance Scoring Methods Employed

Three different methods of relevance scoring were employed in the experiments reported here:

Frequency: Documents are scored using the Cornell variant of the Okapi BM25 weighting function [Singhal et al. 1995; Robertson et al. 1994].

$$w_t = tf_d \times \frac{\log(\frac{N-n+0.5}{n+0.5})}{2 \times (0.25 + 0.75 \times \frac{dl}{avdl}) + tf_d}$$

where w_t is the relevance weight assigned to a document due to term t, tf_d is the number of times t occurs in the document, N is the total number of documents, n is the number of documents

[*] The authors wish to acknowledge that this work was carried out within the Cooperative Research Centre for Advanced Computational Systems established under the Australian Government's Cooperative Research Centres Program.

containing at least one occurrence of t, dl is the length of the document and $avdl$ is the average document length.

Concept: Groups of related terms in a query are called concepts. Documents are scored against each concept and the results are recorded in separate accumulators. The final score s for a document is derived from the concept scores $c_1, \ldots c_n$ using $s = (kc_1 + 1) \times \ldots \times (kc_n + 1)$. In frequency scoring, a document with many occurrences of only one concept may score more highly than another which contains evidence for all the concepts. Concept scoring is designed to boost the weight of documents with evidence for the presence of all concepts.

Distance: Documents are scored using the lexical-distance between instances of concept members as described in [Hawking and Thistlewaite 1996; Hawking et al. 1996]. This method does not require collection frequency statistics.

1.2 Hardware and Software Employed

Since TREC-5, the PArallel Document Retrieval Engine (PADRE) has been completely rewritten to operate on workstations and clusters of workstations. The new PADRE97 software [Hawking 1997b] was used in all experiments reported here. A single-processor Sun Ultra-1 was used except in runs for the VLC track, where a cluster of DEC Alphas was employed.

Interactive query modification was carried out using a new graphical user interface to PADRE97 (`quokka`) which has been designed to facilitate the construction of queries suitable for Concept and Distance as well as Frequency scoring.

1.3 Statistical Testing of Differences Between Runs

Throughout this paper, wherever comparisons are made between pairs of runs, apparent differences between means have been tested for statistical significance using two-tailed t-tests[1] with $\alpha = 0.05$.

2 Automatic Query Generation

Automatic AdHoc, Official Runs `anu6alo1` and `anu6ash1`, semi-official run `anu6avs2` and various unofficial runs.

The goal of experiments using automatic query generation was to provide preliminary answers to the following questions:

1. Using the Frequency scoring method defined above, can the performance of queries be improved by the addition of pseudo-phrases automatically extracted from the query?
2. Using the Frequency scoring method, what is the optimum method for finding and using additional pseudo relevance feedback terms?

Automatic runs were performed for all three official sub-categories: full, description-only, and title-only. In addition, runs were performed using queries derived from title-plus-description. The basic strategy in each case was:

1. generate stems and two-stem phrases from the allowable parts of the topic descriptions;

[1] Future consideration will be given to following the advice of Savoy [1997] who recommended the use of medians rather than means and the use of statistical bootstrapping techniques.

2. score documents against the resulting query; and
3. optionally, update document scores using the additional terms suggested by pseudo relevance feedback.

2.1 Phrases

In PADRE, phrases within documents are identified by computing a `followed-by` proximity relation between the matchsets for all the terms in the phrase.

In generating query phrases, the allowable text of each topic description was converted into a sequence of stemmed non-stopwords and phrase-end markers. A phrase-end marker # was inserted for each SGML tag, for each punctuation mark (except hyphens not surrounded by spaces), for each stopword, and at the end of the topic.

In such token sequences, each contiguous (ordered) pair of stems was considered to be a phrase. Thus, the token sequence `# A B C # D #` would generate the phrases `"A B"` and `"B C"` only. Would-be phrases interrupted only by one of `in`, `to`, `of`, `for`, `on`, or `with` were also accepted and the phrase proximity parameter was increased accordingly when processing documents.

2.2 Relevance Feedback

Subsequent references to *relevance feedback* in fact refer to *pseudo relevance feedback* as there was no human involvement in the feedback process. Instead, highly ranked documents retrieved by an initial query were assumed to be sufficiently relevant as to constitute a useful source of additional query terms.

Robertson [1990] argued that the weights used to select terms to be added to a query should, in general, be different from the document term weights used when processing the query. This approach has been taken here.

2.2.1 Method of Term Selection

Instead of mining complete document text for new terms, only the *hotspots* were mined. A hotspot was defined as a contiguous passage of text within a document which lies within a specified p characters of a term or phrase occurrence. All the hotspots within the T top-ranked documents resulting from running the initial query were mined for new terms. Stopwords and terms from the initial query were not considered. All other terms were stemmed and stored in a hash table and their frequencies of occurrence within the hotspots were accumulated.

Once all hotspots had been mined, selection values for each term in the hash table were computed according to the formula given by Robertson [1990]:

$$a_t = w_t(p_t - q_t)$$

In Robertson's work the p_t and q_t were the probabilities that a relevant and a non-relevant document, respectively, contained the term t. Here, p_t and q_t are the probabilities that any particular term in a hotspot and not in a hotspot, respectively, is the specific term t. That is,

$$p_t = tf_h / l_h$$

and

$$q_t = (tf_C - tf_h)/(l_C - l_h)$$

where tf_h is the frequency of the term in the hotspots, tf_C is the frequency of the term in the whole collection, and l_h and l_C are the number of words in the hotspots and in the complete collection respectively. Robertson's w_t was the relevance weight of the document due to term t but here an approximation was used because hotspots rather than whole documents were examined. Furthermore, in the PADRE97 version employed in the experiments, document frequencies were not stored in the term dictionaries and were thus relatively expensive to compute. Accordingly, the document frequency was estimated from the raw frequency by dividing the latter by 3. For the purpose of term selection, it was assumed that $dl = avdl$, and that $tf_d = 1$, allowing the document term weighting formula to be simplified to:

$$w_t = \frac{\log(\frac{N - tf_C/3+0.5}{tf_C/3+0.5})}{3}$$

2.2.2 Relevance Feedback Training

Relevance feedback in the above-described scheme is controlled by four parameters: T (the number of top-ranking documents to mine for feedback terms, p (the proximity limit defining the extent of the hotspots), n (the number of new terms to add), and w_0 the query term weight to be given to the best new term. A series of experiments over 50 TREC topics (to be described elsewhere) was used to pick a set of values for these parameters.

Table 1: Effectiveness of relevance feedback on past TREC Automatic AdHoc tasks using the feedback parameters ($T = 20; p = 500; n = 30; w_0 = 0.5$). All differences were statistically significant.

	No Feedback	Feedback		
Task	Ave_Prec	Ave_Prec	Precision @ 20	Recall
TREC-3 short	.2063	.3018 (+46%)	+28%	+15%
TREC-4 short	.1925	.2498 (+30%)	+17%	+14%
TREC-5 short	.1502	.1959 (+30%)	+21%	+17%
TREC-3 long	.3441	.3748 (+9%)	+7%	+3%
TREC-5 long	.2356	.2515 (+7%)	+5%	+4%

To confirm the generality of the chosen values, training runs using these parameters were performed on the TREC-3 (short and long), TREC-4, and TREC-5 (short and long) tasks. Results obtained are shown in table 1. In every case, on all three measures, there was a statistically significant benefit from using relevance feedback.

The gain was much smaller for the long topics than for the short. This may seem counterintuitive, as one might expect that better initial queries would yield better text from which to mine relevance feedback terms. However, it is possible that queries derived from the long topic descriptions are closer to optimal, restricting the scope of potential gains from relevance feedback.

Another possibility is that the benefit of relevance feedback terms was scaled down because of higher term frequencies in the longer query. This possibility has yet to be investigated.

2.2.3 Relevance Feedback Failures on Training Topics

The relevance feedback scheme adopted above ($T = 20; p = 500; n = 30; w_0 = 0.5$) produced quite consistent improvement in training. Considering the short topic tasks, on only 23 of the

149 topics did relevance feedback result in loss of more than 0.005 in average precision. Only 5 of the 149 topics were harmed by more than 0.05 and only one by more than 0.10. This topic however saw a loss of 0.47 in average precision! In this case, the unexpanded query achieved an average precision of 0.95. Consequently, additional terms were almost guaranteed to reduce performance.

In earlier experimentation, feedback failures were much more common and consideration was given to the design of a mechanism for turning off relevance feedback on queries which exceeded a threshold of estimated risk. No such mechanism was used in runs reported here.

2.2.4 Parameters Used in Official TREC-6 Runs

Final runs in the very-short, short and long automatic adhoc category were all performed with $(T = 20; p = 500; n = 30; w_0 = 0.75)$ as there was some evidence that a slightly higher value of w_0 might perform better.

2.3 Automatic Adhoc Results

Results for Automatic Adhoc runs are summarised in Tables 2, 3 and 4 and plotted in Figure 1.

Relative to all 57 Category A Automatic Adhoc runs, long-topic run `anu6alo1` retrieved more relevant documents than any other run and achieved best overall precision@20 results. Only the City University title-only run achieved better overall average precision.

The method used in run `anu6alo1` did not perform as well when applied to the description-only and title-only tasks either in absolute terms or relative to all other official submissions in those categories. In these tasks, relative performance was better on the recall rather than on the precision dimension. For example, the unofficial title-only run would have ranked second (of 12) on recall (percentage of all relevant documents retrieved) but eighth on early and average precision.

Title vs. Description: As reported by other groups, the ANU/ACSys title-only run (`anu6avs2`) apparently out-performed the description-only run (`anu6ash1`) by a considerable margin (35% in average precision, 16% in precision @20 and 9% in recall). However, due to large variance in the results, none of these differences was statistically significant ($t(49) = 1.74, 1.21, 1.43$ respectively)!

Title plus Description: A run anu6atd1 using both title and description fields performed 31% better on average precision, 23% better on precision @20 and 16% better on recall than the description-only run. All these differences were statistically significant.

Value of phrases: When all phrases were removed, performance of the `anu6alo1`, `anu6ash1` and `anu6avs2` runs diminished by a small percentage on each of the three measures (average precision, precision@20 and recall). Only in the case of `anu6ash1` precision@20 was the difference statistically significant. Even combining the three runs failed to yield statistically significant differences.

Effectiveness of Relevance Feedback: The effect of relevance feedback in the TREC-6 task is reported in Table 5. As may be seen, feedback produced a statistically significant gain in recall for each of the query sets. The percentage gain in recall for the full-topic queries was similar to that achieved in training on the TREC-3 and TREC-5 long topics (see Table 1). For the short forms of the topic, however, feedback produced much smaller percentage gains in recall than were achieved in training. In contrast to the training results, the apparent improvements in average precision on TREC-6 were not statistically significant.

The possibility that the poorer performance of feedback may have been due to the use of $w_0 = 0.75$ rather than the value of 0.5 used in training was investigated post hoc by re-

Table 2: Average Precision performance of ANU/ACSys Automatic Adhoc runs relative to all official runs in the same category. The number of topics for which the run achieved best (possibly equal best) performance and the number achieving median or better are tabulated in the last two columns. The unofficial title-plus-description run is compared to the group of official description-only runs. There were 50 topics.

Run-id	Category	Mean	Rank	#best	#≥med.
anu6alo1	Full	.2602	1/16	10	49
anu6ash1	Desc. only	.1645	15/29	3	35
anu6avs2	Title only	.2216	8/12*	0	20
anu6atd1	Title/Desc.	.2157	NA	11	41

Table 3: The same runs as in table 2, compared on the basis of overall recall (percentage of all relevant documents retrieved). The figures in parentheses in the #best column show the number of topics for which all relevant documents were retrieved.

Run-id	Category	Percent	Rank	#best	#≥med.
anu6alo1	Full	62%	1/16	14(8)	48
anu6ash1	Desc. only	48%	8/29	11(7)	45
anu6avs2	Title only	55%	2/12	10(7)	30
anu6atd1	Title/Desc.	59%	NA	21(9)	46

Table 4: The same runs as in table 2, compared on the basis of overall recall (percentage of all relevant documents retrieved). Topic-by-topic precision@20 data was not available for all runs.

Run-id	Category	Mean	Rank
anu6alo1	Full	.379	1/16
anu6ash1	Desc. only	.282	8/29
anu6avs2	Title only	.327	8/12
anu6atd1	Title/Desc.	.348	NA

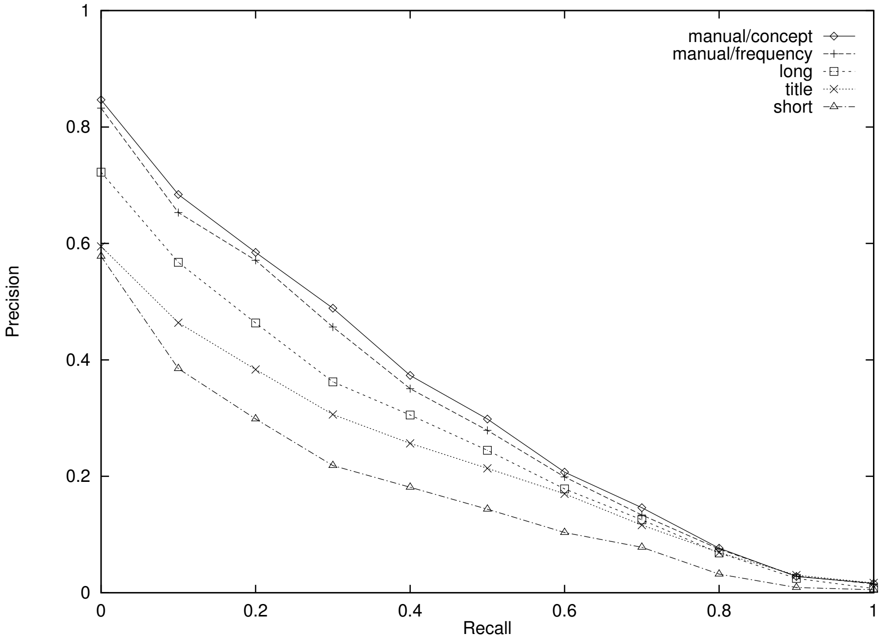

Figure 1: ANU/ACSys Adhoc runs. The two top lines correspond to the same interactively developed set of queries scored using two different methods.

running `anu6ash1` using the lower value of w_0. The new run (`anu6ash3`) achieved slightly better performance than the original (see Table 5) but apparent differences in average and early precision were still not significant. Compared to the training results reported in section 2.2.3, feedback harmed average precision results for a much higher percentage of topics (30% cf. 15% by more than 0.005, 8% cf. 3% by more than 0.05, and 8% cf. 1% by more than 0.10). If feedback could have been selectively switched off for the topics where it did harm, overall average precision would have risen to 0.1867, an increase of 20% over the non-feedback case. This increase is still markedly smaller than that observed in each of the three short-topic training sets, despite the inclusion of feedback failures in the latter results.

2.4 Automatic Adhoc Discussion and Conclusions

Title vs. Description: The text making up the description field of the topic statements appears to be designed to augment rather than to serve as an alternative to the title. For example, the title field of topic 312 contains the highly precise term *Hydroponics* whereas the description *Document will discuss the science of growing plants in water or some substance other than soil* gives an explanation using less precise terms. It is therefore not at all surprising that inclusion of the title terms improved results.

Value of phrases: Despite the lack of a statistically significant result, the use of query phrases as described has led to apparent improvements in nearly every training and official run in which the comparison has been made. There seems to be no reason to discontinue their use.

Value of relevance feedback: The relevance feedback scheme used delivered a consistent apparent benefit (averaged over 50 topics) on each of the tasks to which it was applied. However, its effectiveness on the TREC-6 task would have been much less than had been observed on tests with sets of earlier TREC tasks, even if feedback had been disabled on topics where it did

Table 5: Effectiveness of relevance feedback on TREC6 AdHoc tasks. Feedback parameters ($T = 20; p = 500; n = 30; w_0 = 0.75$) where used in all cases except for run anu6ash3 where $w_0 = 0.5$. Statistically significant differences are marked with an asterisk.

		No Feedback	Feedback		
Run-id	Task	Ave_Prec	Ave_Prec	Prec@20	Recall
anu6avs2	Title-only	.2113	.2216 (+5%)	+11%	+7%
anu6ash1	Description-only	.1556	.1645 (+6%)	+9%	+9%*
anu6ash3	Description-only	.1556	.1680 (+8%)	+8%	+11%*
anu6atd1	Title plus desc.	.2035	.2157 (+6%)	+7%	+7%*
anu6alo1	Full topic	.2461	.2602 (+6%)	1%	+4%*
anu6man1	Manual (Blind)	.2668	.2785 (+4%)	+2%	+4%*
anu6min1	Manual (Interact.)	.3044	.3172 (+4%)	+1%	+3%*

harm.

Contrary to results on earlier TREC tasks, the size of the benefit was similar regardless of the length of the initial query. Table 5 shows that there was a consistent apparent benefit on the three measures (even for manually improved queries). The benefit to recall was statistically significant in six out of the seven cases considered.

The current feedback scheme seems sufficiently robust to justify its routine use, particularly where high recall is important. However, it is hoped that further refinement may result in an adaptive system which reduces harm and magnifies benefits.

3 Manual Query Generation

Manual AdHoc, Official Run anu6min1

3.1 Manual Query Generation Process

A relatively naive user generated a series of manual query sets by successively refining an initial automatically-generated set. In this way it was possible to compare *blind* (no interaction with documents) manual improvements with those obtained after interaction with the test documents. Details of the process were as follows:

Automatic queries were generated from the full topic descriptions using an earlier version of the automatic query generator described above. (The queries were similar but not identical to the queries used in anu6alo1 without feedback.) The topics and queries were then presented to a relatively naive user of the quokka graphical user interface to PADRE. The user was asked to improve the initial queries using any of the following techniques:

1. remove any terms which appeared likely to be distractors,

2. combine any suitable pair of words into a phrase,

3. add new terms which were obviously missing,

4. alter query term weights.

Table 6: Summary of manual runs. The automatic long-topic run feedback `anu6alo1nf` is included as a baseline. None of the runs in this table used relevance feedback. Query processing times are one-observation-only elapsed times observed on a non-dedicated Sun Ultra-1.

Run	Scoring	Ave_Prec	Prec@20	Recall	Time per query (sec.)
`anu6alo1nf`	freq.	0.2461	0.376	2657/4611	20.1
`anu6man1`	freq.	0.2668	0.413	2834/4611	10.1
`anu6con1`	freq.	0.2723	0.427	2929/4611	9.6
`anu6con2`	concept	0.2813	0.438	2980/4611	27.4
`anu6dis1`	dist.	0.0188	0.054	909/4611	276.1
`anu6min1`	freq.	0.3044	0.467	3042/4611	10.4
`anu6min1con`	concept	0.3168	0.486	3099/4611	31.9

Before working on the TREC-6 task, the user was given a training run over the TREC-5 task during which he could compare precision-recall plots for each topic before and after his modifications.

This first phase of manual modification resulted in a set of queries which were used in unofficial run `anu6man1`.

Table 7: Comparison of frequency and concept scoring for two sets of manual queries. Each measure shown is the average of fifty individual topic results. Asterisks indicate statistical significance.

	Frequency Scoring			Concept Scoring		
Run	Ave_Prec	Prec@20	Recall	Ave_Prec	Prec@20	Recall
Blind	0.2723	0.427	0.7429	0.2813(+3%)*	0.438(+3%)*	0.7488(+1%)
Interact.	0.3044	0.467	0.7615	0.3168(+4%)*	0.486(+4%)*	0.7704(+1%)*

Next, the same user was asked to group the terms in the `anuman1` queries into *concepts*. The following explanation of concepts is similar to that given to the user.

> In judging relevance of documents to the topic, "What is the economic impact of recycling tyres?", you might decide that the topic involves three separate concepts: *economic impact*, *recycling* and *tyres* and that relevant documents are likely to contain evidence for the presence of each of them. There may be a whole list of words or phrases which could serve as evidence for the presence of a concept. For example, `profits`, `losses`, and `benefits` might constitute evidence for the presence of *economic impact.*

Sometimes, during this process, new terms suggested themselves. The resulting queries were used in unofficial runs `anu6con1`, `anucon2` and `anudis1`, corresponding to frequency, concept ($k = 30.0$) and distance scoring.

3.2 Concept and Distance Scoring

The present authours have been interested for some time in the idea that queries or sub-queries can be viewed as concept intersections. For a document to be relevant to a topic, there should be evidence for the presence of all of the concepts, not just one. Naturally, the scoring methods must take into account the possibility that evidence actually present may be missed by the query.

ANU/ACSys manual queries in TREC-4 and TREC-5 were scored using the lexical length of concept spans [Hawking and Thistlewaite 1996], but it was recognised that span-based queries were harder to generate. In TREC-5, efforts were made to use automatic methods to augment manually generated distance queries. These efforts were moderately successful and could have been more so had they been improved interactively.

An unfortunate aspect of distance-based scoring is that errors in defining concepts, such as placing synonyms in different concepts, may dramatically alter the performance of the query. The present work proposes *Concept* scoring (defined in Section 1.1) as a method with the potential to gain benefit from concept intersections without the degree of risk associated with distance scoring. Using the method, each concept was scored using the Frequency function as an independent sub-query and the resulting scores combined in a way which boosted the overall scores of documents containing more of the concepts.

An effort was made to compare the benefits of this concept scoring compared to span-scoring. However, there was insufficient time to test the new PADRE97 implementation of spans prior to use and results obtained may have been affected by coding bugs.

3.3 Interactive Manual

In the final stage of manual query refinement, the same user was allowed to interactively modify the concept queries by running them and scanning the documents retrieved. Unfortunately, due to a misunderstanding, this interaction was done over CD2/CD5 rather than CD4/CD5. This resulted in a new set of queries (sometimes using negative query term weights) which were subsequently re-run over CD4/CD5 as official run `anumin1`.

3.4 Manual Adhoc Results

Table 6 summarises the manual adhoc runs. Blind manual tweaking (including organisation into concepts and use of concept scoring) of the initial queries not only produced statistically significant benefits in average precision (+14%), precision @20 (+16%) and recall (+7%) but also halved the running time of the queries. By comparison, automatic feedback applied to the initial queries gained less than the manual tweaking and took six times as long to run.

Concept scoring worked significantly better than Frequency scoring in both of the cases shown in Table 7. The benefit is most evident in the precision rather than the recall dimension.

Distance scoring worked very poorly as shown by the results for run `anu6dis1` in table 6.

Interactive modification of queries produced statistically significant benefits in average precision (+12%) and early precision (+9%) despite the interaction using incorrect data. The apparent improvement of 3% in recall was not significant.

3.5 Manual Adhoc Discussion and Conclusions

The fact that a non-expert user was able to substantially improve both the speed and the effectiveness of good automatic queries in a short time (even without interaction with the documents) indicates that there is considerable scope for improvement in the automatic query generation process.

In the future, consideration will be given to using automatic queries generated from topic titles only as the starting point for manual runs, particularly in time-limited situations such as the High-Precision track. It has been noted that non-feedback automatic queries generated from only the topic title found an average of 5.9 relevant documents in the first 20 retrieved compared with 7.5 for the corresponding long-topic versions, but took an average of only 1.3 seconds to process, compared with 20.1 seconds.

It is notable that both concept scoring and relevance feedback are (independently) capable of significantly improving the performance of Okapi-scored interactively developed queries. The results for concept scoring are encouraging but further work is required to confirm generality and to hopefully improve the method.

At the time of writing, it had not been determined whether the disappointing results for Distance scoring arose from bugs in the code, from poor construction of the concept groups, from poor automatic generation of the `span` commands used in scoring or for some other reason. Further work is needed to investigate and hopefully to rectify the cause of the poor performance and to further compare the three alternative scoring methods.

4 Filtering Experiments

Filtering Track, Official Runs `anu6fltU1` and `anu6fltU2`

Filtering queries were generated from topic descriptions and training judgments, using the programs `topic-aqg` and `freq-aqg`, and applied to the training collection to derive relevance score *thresholds*. Queries and thresholds were then applied to the test collection to generate the TREC-6 submissions.

The `topic-aqg` program was used to extract terms and two word phrases from topic descriptions, using methods similar to those used in the Adhoc tasks. Terms were weighted more highly if they appeared more than once, if they were written in capitals and if they appeared in the topic title. The resulting terms and phrases were ordered by decreasing weight.

The `freq-aqg` program was used to extract terms and two word phrases from training documents. Each term and phrase from the training documents was ranked and weighted according to to $P_r - P_i + 1$ where P_r is the probability of it occuring in a relevant training document and P_i was the corresponding probability for irrelevant training documents.

To generate a filtering query, the best n terms/phrases were taken from the `topic-aqg` ranking and the best m terms/phrases from the `freq-aqg` ranking (duplicates were not removed), and term weights from these sources were scaled by a factor of w_n and w_m respectively.

To assist in finding optimal query generation parameters, the Generalized Reduced Gradient (GRG2)[2] nonlinear optimization algorithm was employed. The decision variables were n, m, w_n and w_m, and the objective function was average utility F_1 across all topics when run on the training collection. As n and m increased, the utility tended to gradually but uniformly increase, so large values of n and m were chosen (n=80 and m=80). It was also found that values of $w_m \approx 1.1904$ and $w_n \approx 0.7317$ would achieve greater utility than other choices of scaling factors (7.5% greater utility than for $w_m = 1$ and $w_n = 1$). These optimal training-collection values were used in test-collection query generation[3].

Documents were scored according to the Okapi variant used in Adhoc runs, but document frequencies and collection sizes were taken from the training collection rather than the test collection. Thresholds for each topic were set at the PADRE document score cutoff corresponding

[2] Developed by Leon Lasdon, University of Texas at Austin, and Allan Warren, Cleveland State University and implemented in Microsoft Excel 97.

[3] It would be interesting to optimise over the test judgments now that they are available and to compare the resulting parameter values with those actually used.

to maximum utility on the training collection. Test documents failing to reach the threshold score were rejected.

4.1 Filtering Results

Table 8: Summary of the performance of ANU/ACSys Filtering runs. There were 47 topics. *Num_ret* is the average number of documents returned by the run while *Num_rel* is the average total number of relevant documents. *Zeros* shows the number of topics for which the run returned zero documents and (in parentheses) the number of topics for which a group best result was achieved by the run while returning zero documents.

Run-id	Ut. Function	Utility	Rank	#best	#≥med.	Num_ret	Num_rel	Zeros
anu6fltU1	F1 (prec.)	12.97	6/17	7	36	35	146	12(3)
anu6fltU2	F2 (recall)	-57.55	5/17	2	37	89	146	3(1)

Table 8 summarises the performance of the ANU/ACSys filtering runs. Performance was quite pleasing. Only two groups achieved better results on the F2 measure and three on F1.

The number of documents returned by the ANU/ACSys runs was, on average, much less than the total number of relevant documents, even in the F2 case. Note that returning zero documents achieves an F1 score of 0.0 but an F2 score of -146 (on average).

4.2 Filtering Discussion and Conclusions

Future work on the same basic filtering approach is likely to investigate:

- optimisation over more inputs (stemming/nonstemming, phrases/notphrases, different weighting profiles etc.);
- use of stemming;
- separate optimisation for F1 and F2;
- use of $n > 80$ and $m > 80$; and
- different term ranking and term weighting strategies.

5 Experiments with a Larger Collection

Very Large Collection Track, Official Runs anu6vlb1 and anu6vlc1

The main goal of research in this area was to design, implement and test a scalable retrieval architecture. In the recent re-design of PADRE, attention was paid to minimising communication, minimising synchronisation points and maximising use of communication buffering. The bulk of this work has been reported elsewhere [Hawking 1997b; Hawking 1997a].

Figure 2 predicts the scalability of PADRE97 query processing for three different hardware environments. It suggests that as data size grows, query processing speed on a workstation will deteriorate approximately linearly with data size until physical memory limits cause paging and consequent more rapid degradation. By contrast, if the number of workstations in a cluster is

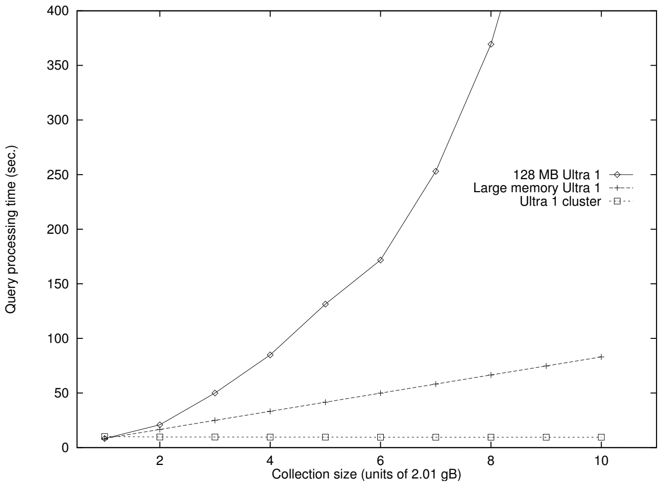

Figure 2: Elapsed query processing times for processing collection sizes measured in 2.01 gigabyte units on three different systems: Ultra 1 (observed times); Ultra 1 with memory hypothetically increased in proportion to the data size (projected times); a cluster of Ultra 1s with one search engine for each unit of collection size (scaled up from smaller data on SPARC-based Fujitsu parallel machine). (Reproduced from [Hawking 1997b].)

increased in proportion to the increase in data size and the data is evenly distributed across the cluster, then query processing times need not increase at all.

Hawking [1997b] also reported results for query processing over a 102 gigabyte (replicated) collection using ten 64-megabyte SGI workstations in a student laboratory.

Further work is needed to determine how scalability is affected by network latency when query processing times are short.

Unfortunately, the SGI laboratory was only available for a few days during student vacation and a VLC run conducted during that time was affected by a bug. Official runs in the VLC track were therefore processed on a cluster of eight DEC Alphas. Using eight nodes to process ten times as much data suggests that a VLC/baseline query processing time ratio of 1.25 would be an appropriate target. Achievement of this target assumes that times on each Alpha would increase by 25% in response to a 25% increase in data size and requires perfect load balance across nodes (unlikely to be achieved in practice).

Optimised queries were used (in which only the 15 lowest-frequency terms were processed) and only the top 20 documents were retrieved. Retrieving 1000 documents with the same queries increased the average time by 23% for the baseline. Query processing full (average 30 term) queries over the baseline retrieving 1000 documents took 217% longer. The initial queries used in both cases were generated from the full topic descriptions by an earlier version of the automatic query generator. They are thus similar to `anualo1nf`.

5.1 VLC Results

Table 9 shows the VLC measures taken from the two ANU runs in the VLC task. Results for all groups are presented in the VLC Track Overview elsewhere in these proceedings.

Of the seven official 20-gigabyte runs, the ANU/ACSys run:

1. required the least disk space (due to effective compression) despite full term-position in-

Table 9: Summary of ANU/ACSys VLC runs on a Alpha Farm consisting of eight 266 MHz EV5 Alphas connected by both 155 Mbit/sec ATM and switched 10BaseT. The only disk storage local to the Farm was a 20 gB RAID array connected by SCSI to one of the Alphas. Six of the Alphas (including the one supporting the RAID array) were equipped with 128 MB of RAM and the remaining two with 192 MB. Each Alpha features on-chip primary and secondary caches and an off-chip 2MB cache. The baseline run was carried out on the Alpha with directly connected RAID array. Indexing of the VLC was carried out sequentially using only the RAID-equipped node. The VLC query processing runs were carried out using 8 of the nodes. The RAID node ran the UIF process and also searched a small (1.1 gB collection). Disk space figures quoted exclude as-supplied compressed text (baseline: 0.80 gB vlc: 8.00 gB) but include all data structures generated, whether used in query processing or not. Costs were obtained from the Digital Equipment Corporation website on 15 September, 1997 and include the cost of the nodes and the RAID storage array. The VLC cost (only) includes the cost of the ATM switch connecting the nodes.

	`anu6vlb1`	`anu6vlc1`	
Measure	Baseline	VLC	VLC/Baseline
Precision@20	0.356	0.509	1.43
Ave. query processing time	12.1 sec.	50.5 sec.	4.17
Data struct. bld. time	1.405 hr.	15.6 hr.	11.1
Disk space	0.626 gB	6.06	9.68
Memory	128 MB	1152 MB	9
H/W cost (USD)	23.9	95.1	3.98
gB-queries/hour/kilo$	25.9	14.9	0.598

formation being included in indexes.

2. achieved second-best scalability of query processing time. However, the VLC/baseline ratio of 4.17 was much higher than the target (1.25).

3. achieved the third fastest indexing rate, despite using only a single processor. The two faster runs each used four or more processors.

4. achieved the third fastest query-processing rate. However, query processing was 20-40 times slower than the two faster runs.

5. ranked fourth on the bang-per-buck measure but was a factor of nearly 500 behind the best-scoring system.

Like all other groups, ANU/ACSys observed a large increase in early precision from the baseline to the VLC run. Comparison of actual precision values is not particularly meaningful because of a diversity of query construction methods used. The three groups (ANU, ATT and City) which derived queries from the full topic text achieved similar early precision on the VLC (0.509, 0.530 and 0.519 respectively). Groups which used only the short topic statement performed significantly worse and manually constructed queries performed significantly better.

5.2 VLC Discussion and Conclusions

The failure to more closely approach the VLC/Baseline query processing time ratio of 1.25 was almost certainly because disk storage was centralised on one of the nodes rather than distributed.

Improvement in query-processing rate may be achievable through application of some of the following additional optimisations:

1. Limit the number of document accumulators (and continue to processing terms in order of increasing *df*). This should improve memory reference locality and dramatically reduce the cost of the ranking sort.

2. Arrange the inverted file indexes in order of increasing *df* to maximise memory residency of the compressed index entries.

3. Improve the scalability ratio by using distributed disks rather than a centralised RAID box and ensuring good load balance.

4. Improve the queries. The best query processing rate among the seven runs was achieved using short, high-quality (manually generated) queries.

5. Study the relative costs of index entry decompression and disk I/O. If the former is expensive relative to the latter, query processing may be speeded up by using uncompressed or partly compressed indexes.

6. Remove term position information from the indexes. This information was not used in processing either the VLC or baseline queries. Removing it could reduce memory demands and dramatically speed up decompression of postings lists.

Improvement in the bang-per-buck measure will result from improvement in query-processing speed and/or from the use of cheaper hardware. It is interesting to note that the proposed use of local disks rather than a centralised RAID on the cluster of Alphas would reduce rather than increase the cost of the system. It is likely that the use of collection-selection techniques could dramatically improve bang-per-back performance without significantly harming early precision on the large collection.

The indexing rate of 1.29 gB/elapsed hour achieved on the VLC using a single 128 MB workstation is quite satisfactory given that the input text remains in compressed form, that the index contains full position information and that total disk space requirements (including temporary files) only amount to one third of the raw text size. With local disks on each of the eight Alpha nodes it should be possible to increase the indexing rate by close to a factor of eight, depending upon degree of load balance achievable, to over 10 gigabytes/elapsed hour.

It is planned to investigate the reason for the higher early precision observed with the larger collection in future work.

6 Conclusions

The pseudo relevance feedback method proposed here has been shown to produce consistent average benefit for all the sets of topics on which it has been tried. However, the benefit gained on the TREC-6 Adhoc tasks was not as great as that observed in training with earlier TREC tasks.

Combined with a Cornell variant of City University's Okapi BM25 scoring function, this feedback method was used very successfully in the Long-topic Automatic Adhoc category, achieving best average precision. The same run achieved best recall and best precision@20 of all 57 official Automatic Adhoc runs and was surpassed by only one run on average precision.

The same basic automatic method was used as the basis for successful submissions in the Manual Adhoc, Filtering and VLC tasks. Starting from automatically generated queries, a relatively naive PADRE97 user was able to achieve third best results in the Manual Adhoc category with only a relatively small investment of time, despite interacting with only part of the TREC-6 data set. The effectiveness of manual refinement indicates that there is scope for improvement in the automatic query generation process. In Filtering, a Generalised Reduced Gradient non-linear optimisation method was used to set score thresholds. Only two groups achieved better official results on the F2 utility measure and three on F1.

A new BM25-based method of Concept scoring was shown to produce a small but significant gain in precision on the Manual Adhoc task. Further work is needed to prove (and hopefully improve) its usefulness. The automatic generation of concepts suitable for use in Concept and Distance scoring remains a goal of future research.

Bibliography

Hawking, D. 1997a. PADRE for COWs. In P. Mackerras Ed., *Proceedings of the Sixth Parallel Computing Workshop, PCW97* (Canberra, September 1997). Department of Computer Science, The Australian National University, Canberra. Paper P1-B.

Hawking, D. 1997b. Scalable text retrieval for large digital libraries. In C. Peters and C. Thanos Eds., *Proceedings of the First European Conference on Digital Libraries*, Volume 1324 of *Lecture Notes in Computer Science* (Pisa, Italy, September 1997), pp. 127–146. Springer, Berlin.

Hawking, D. and Thistlewaite, P. 1996. Relevance weighting using distance between term occurrences. Technical Report TR-CS-96-08, Department of Computer Science, The Australian National University, `http://cs.anu.edu.au/techreports/1996/index.html`.

Hawking, D., Thistlewaite, P., and Bailey, P. 1996. ANU/ACSys TREC-5 experiments. In E. M. Voorhees and D. K. Harman Eds., *Proceedings of TREC-5* (Gaithersburg MD, November 1996), pp. 359–376. NIST special publication 500-238, `http://trec.nist.gov`.

Robertson, S. E. 1990. On term selection for query expansion. *Journal of Documentation 46*, 4, 359–364.

Robertson, S. E., Walker, S., Hancock-Beaulieu, M., and Gatford, M. 1994. Okapi at TREC-3. In D. K. Harman Ed., *Proceedings of TREC-3* (Gaithersburg MD, November 1994). NIST special publication 500-225.

Savoy, J. 1997. Statistical inference in retrieval effectiveness evaluation. *Information Processing and Management 33*, 4, 495–512.

Singhal, A., Salton, G., Mitra, M., and Buckley, C. 1995. Document length normalization. Technical Report TR95-1529, Department of Computer Science, Cornell University, Ithaca NY.