ACSys TREC-8 Experiments

David Hawking*
CSIRO Mathematics and Information Sciences,
Canberra, Australia
Peter Bailey and Nick Craswell
Department of Computer Science, ANU
Canberra, Australia
David.Hawking@cmis.csiro.au, {peterb,nick}@cs.anu.edu.au

October 12, 1999

Abstract

Experiments relating to TREC-8 Ad Hoc, Web Track (Large and Small) and Query Track tasks are described and results reported. Due to time constraints, only minimal effort was put into Ad Hoc and Query Track participation. In the Web Track, Google-style PageRanks were calculated for all 18.5 million pages in the VLC2 collection and for the 0.25 million pages in the WT2g collection. Various combinations of content score and PageRank produced no benefit for TREC style ad hoc retrieval. A major goal in the Web Track was to make engineering improvements to permit indexing of the 100 gigabyte collection and subsequent query processing using a single PC. A secondary goal was to achieve last year's performance (obtained with eight DEC Alphas) with less recourse to effectiveness-harming optimisations. The main goal was achieved and indexing times are comparable to last year's. However, effectiveness results were worse relative to last year and query processing times were approximately double.

1 Introduction

The work reported here comprises a number of text retrieval experiments conducted within the framework of TREC-8 and addressing questions of interest in the following research areas: Practical information retrieval; Exploitation of link information.

ACSys completed Automatic Adhoc, Query Track, Large Web and Small Web tasks.

1.1 Basic Relevance Scoring Method

As in TREC-6 and TREC-7 [Hawking et al. 1997], the basic relevance scoring method used in official ACSys adhoc runs was the Cornell variant of the Okapi BM25 weighting function [Singhal et al. 1995; Robertson et al. 1994]

$$w_t = q_t \times tf_d \times \frac{\log(\frac{N-n+0.5}{n+0.5})}{2 \times (0.25 + 0.75 \times \frac{dl}{avdl}) + tf_d} \tag{1}$$

*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.

where w_t is the relevance weight assigned to a document due to query term t, q_t is the weight attached to the term by the query, 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 containing at least one occurrence of t, dl is the length of the document and $avdl$ is the average document length (measured either in bytes or in indexable words).

1.2 Hardware and Software Employed

Two different versions of PADRE software, known as PADRE98 and PADRE99 were used in experiments reported here. An Intel PC with 1 gB of RAM and the Linux operating system was used throughout.

1.3 Query Expansion

A set of synonyms derived by manual inspection of 150 past topics was used in some of the runs with PADRE99.

Relevance feedback, as described in [Hawking et al. 1997] was used in the PADRE98 Ad Hoc runs. A more efficient implementation of the same model was used in some of the PADRE99 runs.

1.4 Retrieval in a Production Environment

Many users of Web search engines pose short queries in which small lexical differences in documents and queries are far more significant than they are with long TREC topics. Some of these tiny but potentially vital lexical signals may be eliminated by operations which normally contribute to successful TREC ad hoc participation: stemming, stopword elimination and case folding. Examples of queries in which such differences may be important include: `"the Pope"`, `"to be or not to be"`, `"new Apples"`, and `Hawking`.

We hypothesise that stemming, stopword elimination and case folding operations should be applied (when appropriate) during query processing rather than during indexing. The PADRE99 runs reported here relate to indexes which, although case folded, are unstemmed and include all words, numbers and letter-digit combinations.

Another feature of everyday Web search engine usage is that users who enter short queries tend to be dissatisfied when results at the top of the ranking do not contain all of the query terms. For queries of less than five words, the PADRE99 software always presents documents containing more of the query terms ahead of those with less.

1.5 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 with $\alpha = 0.05$.

2 Automatic AdHoc Runs

In the following the codes T, D and N are used to indicate use of Title, Description and Narrative fields of the Ad Hoc topic statements. TDN implies use of all three fields.

Figure 1 reports results for four PADRE98 runs made using the same methods used by ACSys in TREC-7. Almost identical source code was used but two changes were made to overcome problems encountered in TREC-7: The maximum length of a word recorded in the index was increased from 12 to 16 characters and the stop word list was considerably shortened.

Comparing feedback and no-feedback versions of the PADRE98 TDN runs, differences in average precision, precision at 20 documents retrieved, and recall were all statistically significant (+11%, +6%, +7% respectively).

Table 1: Performance of ANU/ACSys Automatic Adhoc runs using PADRE98 software and TREC-7 methods. Stopwords were not included in the index. (The stopword list included 88 stems.) Index words were stemmed. Query term weights were assigned on the basis of frequency within previous queries. Pseudo-phrases (ie. word pairs) were automatically generated for TD and TDN and concept scoring ($k = 1$) for the T runs. Relevance feedback used the top 20 new terms derived from hotspots (defined as the text within 500 characters of a query word occurrence) in the top 20 documents found by the original query. The stopword list included 88 stems.

Run-id	Topic Fields	Ave Prec	P@20	Recall	Notes
acsys8alo	TDN	.2935	.4400	.7504	
acsys8alo2	TDN	.2637	.4160	.7041	No relevance feedback
acsys8amo	TD	.2792	.4260	.7524	Unofficial run, 3 Aug 99
acsys8aso	T	.2740	.4280	.6978	Unofficial run, 3 Aug 99

Table 2: Performance of ANU/ACSys Automatic Adhoc runs using PADRE99 software. Stopwords were not excluded from the index and index words were unstemmed. Query term weights were just the occurrence frequency within the current query. Relevance feedback used the top 30 new terms derived from hotspots (defined as the text within 100 indexable words of a query word occurrence) in the top 20 documents found by the original query. The best feedback term received 0.75 of the query-term weight assigned to a query term which occured only once in the initial query.

Run-id	Topic Fields	Ave Prec	P@20	Recall	Notes
acsys8aln2	TDN	.2560	.4060	.6955	Corresponds to acsys8alo2
acsys8amn	TD	.2353	.3790	.6187	Synonym expansion and relevance feedback
acsys8asn	T	.2309	.3790	.5931	Synonym expansion and relevance feedback

Comparing the three feedback runs, the only significant differences on any of the measures betwen T, TD and TDN runs are the recall differences between T and each of the longer-topic runs (+8% in both cases).

Queries used in the `acsys8aln2` run were the ones generated for `acsys8alo2`, but translated into the PADRE99 query language. A stem-matching operator was appended to each literal in the `acsys8alo2` queries to ensure that the effect of stemming was achieved despite the fact that PADRE99 indexes were unstemmed.

There was no statistical difference on any of the three effectiveness measures between the two runs. However, the `acsys8aln2` queries required an average of 12.98 sec. to run compared with 4.29 sec. per query for `acsys8alo2`, on the same hardware.

Query-time stemming is slower because a potentially large chunk of the term dictionary must be scanned for terms which stem to the target and multiple postings lists accessed instead of just one. The problem is worse than might be imagined because there are often a surprising number of "words" (including misspellings) which share a common stem, according to rule-based stemming (Porter method).

2.1 Follow-up Runs

Further runs were conducted post-hoc to determine the effectiveness of relevance feedback and synonym expansion as used in the PADRE99 runs. Results are tabulated in Figure 1 but have not yet been statistically analysed. On the surface, it appears that synonym expansion (as implemented) had negligible effect and that relevance feedback (as implemented) was clearly beneficial. Increasing the amount of topic text appears to improve performance but the benefit of relevance feedback appears to diminish as the length of the topic text increases, particularly on the precision dimension.

3 Query Track

A set of short queries was generated manually by David Hawking for contribution to the track pool. Hastily constructed `perl` scripts were used to translate the sets of queries into a form suitable for processing by PADRE99. No stemming was applied, no stopwords were eliminated and no particular smarts (forgive the pun) were applied during processing.

4 Small Web Task

The major questions addressed by this track were:

1. Are the best methods for retrieval over the ad hoc data also the best for the WT2g collection?
2. Can link information be used to enhance retrieval?

The ACSys contribution to answering the first question was to run the `acsys8mn` query set over the WT2g data using the identical processing parameters as had been used in the Ad Hoc track. The resulting run was called `acsys8wm`. An answer to the question can only arise from a study of the collection of runs.

ACSys contributed to the second question by combining PageRank [Brin and Page 1998] scores with the content scores generated by the `acsys8wm` run. For each topic, each document's content score was normalised so that the highest-scoring document scored 1.0. PageRank scores (which are topic independent) were scaled relative to the highest PageRank score. Normalised content and PageRank scores were treated as orthogonal axes and each document was represented as a point in 2-space. The document's final score was taken as the vector distance of that point from the origin.

Cursory inspection indicated that differences between the baseline rankings and rankings obtained by this means were small. Accordingly, in some experiments reported below, the normalised PageRank scores were multiplied by a factor of 10.0 in order to both create an effect large enough to measure and to increase the chance that pages with high PageRank scores would be judged.

4.1 How were PageRanks Computed?

PageRank is a measure of "link popularity" within a set of hyper-text documents. One way to understand the concept of link popularity is to assume a "random surfer" is walking the graph of Web pages in question, following hyper-links at random. Specifically the surfer has the following behaviour:

1. The surfer has some bookmarks, a subset of the available pages. (In the experiments reported here the complete set of available pages constituted the bookmarks except in one case where only the root pages of each of the 953 servers was bookmarked.)
2. The surfer picks a random page from the bookmarks and visits it.
3. If the visited page has no links to other pages, go to step 2

Results tabulated on Wed Oct 6 15:55:18 1999 on peace.anu.edu.au

Average Precision

	plain	-rf	-syn	rf/syn	average
short	0.1976	0.2317	0.1951	0.2310	0.2139
medium	0.2124	0.2324	0.2134	0.2341	0.2231
long	0.2209	0.2422	0.2241	0.2404	0.2319
average	0.2103	0.2354	0.2109	0.2352	

Precision @ 20

	plain	-rf	-syn	rf/syn	average
short	0.3520	0.3900	0.3490	0.3800	0.3678
medium	0.3800	0.3930	0.3790	0.3780	0.3825
long	0.3910	0.4160	0.3790	0.4000	0.3965
average	0.3743	0.3997	0.369	0.386	

Topic-by-topic recall

	plain	-rf	-syn	rf/syn	average
short	0.5361	0.5977	0.5288	0.5931	0.5639
medium	0.5691	0.6171	0.5686	0.6132	0.592
long	0.5903	0.6311	0.5852	0.6232	0.6075
average	0.5652	0.6153	0.5609	0.6098	

Figure 1: Follow-up runs with PADRE99 on the Automatic Ad Hoc task, exploring the effectiveness of synonym expansion and relevance feedback for each topic length.

4. Otherwise, pick a link at random, visit it and go to step 3.

The PageRank of a page is the probability that the surfer will be visiting that page at any point in time.

In a simple system with two pages that link to each other, the probability that a surfer will be at one page is 0.5. In a more complex system, the probability that the surfer will be at a page is roughly proportional to the in-degree of that page. If a page has no incoming links and is not on the bookmarks, the probability of a visit (and hence the PageRank) is zero. If every page has two links, but one link from every page points to page A, page A will have a very high PageRank, and the page pointed to by A will also inherit a high PageRank.

PageRanks were calculated using the iterative methods suggested in [Page et al. 1998]. Real Web users are not random surfers. They are more selective about link-following, they use the Back button to revisit pages and may find new pages through searching rather than browsing. However PageRanks are an indication of likely page popularity. A page with many incoming links and high PageRank, like `www.microsoft.com` is more likely to be visited than one with few incoming links and low PageRank, like `pastime.anu.edu.au`. A page with no incoming links and zero PageRank is very unlikely to be found. Search engines rely on "spiders" to crawl the Web, and these too are less likely to find a page if it has fewer incoming links [Lawrence and Giles 1999]. For these reasons, a searching/browsing user is more likely to find a page with more incoming links. However, "popularity" and document utility/relevance, as measured in ad hoc retrieval tasks such as those in TREC, may well be orthogonal.

4.2 Small Web Results

Table 3: Performance of ANU/ACSys Small Web runs using PADRE99 software. Except for the use of PageRank scores, conditions were identical to those prevailing in the Ad Hoc task.

Run-id	Topic Fields	Ave Prec	P@20	Recall	Notes
acsys8wm	TD	.3009	.387	.8231	Content-only
acsys8wmp	TD	.3007	.387	.8213	PageRank wt = 1.0
acsys8wmq	TD	.2804	.3700	.8025	PageRank wt = 10.0
acsys8wmr	TD	.3007	.387	.8213	PageRank wt = 1.0 server bookmarks

Results of Small Web runs are summarised in Table 3.

The minuscule difference in average precision between `acsys8wm` and `acsys8wmp` results from a large number of topics in which PageRanks make no difference at all, and a very small number where they cause harm. The run `acsys8wmr` which used Server bookmarks performed very similarly.

When the normalised PageRanks were scaled up by a factor of 10.0 (run `acsys8wmq`), all three measures were significantly depressed relative to the baseline (by -7%, -4%, and -3% for average precision, precision at 20 and recall). Only topic 413 ("steel production") benefits from use of PageRanks:

```
TOPIC      AVE PREC.              P@20                  RECALL
413  (0.0738 0.0866, +17\%) (0.1500 0.2000, +33\%) (0.7500 0.7500,  +0\%)
```

All other topics were either unaffected or were adversely affected.

Further runs were conducted post-hoc to determine the effectiveness of PADRE99 relevance feedback and synonym expansion as applied to the Small Web data. Results are tabulated in Figure 2 but have not yet been statistically analysed. On the surface, results are quite contrary to those obtained on the Ad Hoc collection. Increasing topic length still improves performance but synonym expansion (as implemented)

Average Precision

	plain	-rf	-syn	rf/syn	average
short	0.2678	0.2736	0.3262	0.3242	0.298
medium	0.2999	0.3178	0.3530	0.3525	0.3308
long	0.3117	0.3222	0.3701	0.3663	0.3426
average	0.2931	0.3045	0.3498	0.3477	

Precision @ 20

	plain	-rf	-syn	rf/syn	average
short	0.4890	0.4910	0.5180	0.5150	0.5033
medium	0.5480	0.5530	0.5740	0.5640	0.5598
long	0.5690	0.5580	0.6020	0.5790	0.577
average	0.5353	0.534	0.5647	0.5527	

Topic-by-topic recall

	plain	-rf	-syn	rf/syn	average
short	0.7085	0.7257	0.7931	0.7806	0.752
medium	0.7428	0.7693	0.7976	0.8207	0.7826
long	0.7575	0.7785	0.8093	0.8244	0.7924
average	0.7363	0.7578	0.8	0.8086	

Figure 2: Pre-deadline runs with PADRE99 on the Small Web Task, exploring the effectiveness of synonym expansion and relevance feedback for each topic length

was clearly beneficial while relevance feedback was almost useless by itself and harmful when combined with synonym expansion.

5 Large Web Task

PADRE99 includes various engineering improvements to reduce the impact of indexing and query processing on the virtual memory system. These were sufficient to allow indexing of the 18.5 million page, 100 gigabyte VLC2 collection in under 10 hours on a single 450MHz Pentium 3 system with 1gB of RAM. The official runs relate to an index built as eleven separate components covering approximately 9 gigabytes of text each. The index included all unstemmed words comprising letters only up to a maximum of 12 characters. Each posting recorded the *tf* value for a term,document pair. Position information was not included.

Subsequently, the data was re-indexed in four chunks of approximately 25 gigabytes each.

Each of the 10,000 Large Web Task queries was processed as follows:

1. Stopwords from a list of 51 were eliminated. Note that words were not stemmed.
2. The remaining query words were then sorted by increasing *df*, as estimated by the first index component.
3. Terms were processed until the end of the query unless a high frequency word (occuring in more than 5% of documents) was encountered after at least three terms had been processed.
4. Document content-scores were accumulated in a hash-addressed set of accumulators. No more than 100,000 accumulators were permitted to become active.

5.1 Large Web Task Results

Timing results are reported in detail in the Web Track Overview. In general, query processing speed was acceptable at under about 4 seconds elapsed time per query, whether or not PageRanks were used. However, effectiveness was relatively poor, due perhaps to the use of individual words rather than stems (which may bias term *df* weighting as well as failing to discover useful matches) and to the use of too few document accumulators[1] without a sensible ordering of postings within postings lists (as was done last year.) Time did not permit much experimentation or testing.

Table 4: Performance of ANU/ACSys Small Web runs using PADRE99 software. Except for the use of PageRank scores, conditions were identical to those prevailing in the Ad Hoc task. PageRank scores were derived using Universal bookmarks (i.e. every page was bookmarked).

Run-id	Topic Fields	Mod. Ave Prec	P@10	P@20	Notes
`acsys8lw0`	TD	.2352	.3440	.3360	Content-only
`acsys8lw0_pr1`	TD	.2363	.3460	.3360	PageRank wt = 1.0
`acsys8lw0_pr10`	TD	.2231	.3380	.3350	PageRank wt = 10.0

Results of Large Web runs are summarised in Table 4. It is interesting that the measures for modified average precision and for P@10 are numerically higher for the equal-weighted PageRank run although it is not expected that the differences are statistically significant.

[1] Once all the available relevance scoring accumulators have been assigned to documents, score contributions for other documents are ignored.

6 Conclusions

The lack of significant difference between the PADRE98 and PADRE99 long topic Automatic Ad Hoc runs suggests that an index built without stopword elimination and without stemming can be used to achieve the same query processing effectiveness, while avoiding loss of potentially useful information. The significant increase in query processing cost is something which needs to be addressed, as is the relatively poor performance of the efficient relevance feedback mechanism.

The incorporation of PageRank scores in rankings in the Large and Small Web tasks produced no benefit. It is concluded that PageRanks are not useful within the TREC context, even when using queries actually taken from Web search engine logs.

The results on the Large Web task indicate that it is quite feasible to index a 100 gigabyte collection of Web documents on a US$7,000 PC and to process queries at a reasonable rate. The compactness of PADRE99 indexes has permitted the demonstration of query processing over the 100 gigabyte collection using a Dell laptop (266MHz Pentium II processor with 128 MB of RAM) and only 6.5 gB of disk space. Unfortunately, to achieve this has required taking query processing short cuts (avoiding phrases, stemming and query expansion and limiting the number of document accumulators) which cause harm to effectiveness.

Acknowledgements

Bibliography

BRIN, S. AND PAGE, L. 1998. The anatomy of a large-scale hypertextual Web search engine. In H. ASHMAN AND P. THISTLEWAITE Eds., *Proceedings of the Seventh International World Wide Web Conference*, Volume 30 of *Computer Networks and ISDN Systems. The International Journal of Computer and Telecommunications Networking* (Amsterdam, April 1998), pp. 107–117. Elsevier. Brisbane, Australia.

HAWKING, D., THISTLEWAITE, P., AND CRASWELL, N. 1997. ANU/ACSys TREC-6 experiments. In E. M. VOORHEES AND D. K. HARMAN Eds., *Proceedings of the Sixth Text Retrieval Conference (TREC-6)* (Gaithersburg MD, November 1997), pp. 275–290. U.S. National Institute of Standards and Technology. NIST special publication 500-240.

LAWRENCE, S. AND GILES, C. L. 1999. Accessibility of information on the web. *Nature 400*, 107–109.

PAGE, L., BRIN, S., MOTWANI, R., AND WINOGRAD, T. 1998. The pagerank citation ranking: Bringing order to the web. Technical report (January), Stanford, Santa Barbara, CA 93106. `http://www-db.stanford.edu/~backrub/pageranksub.ps`.

ROBERTSON, S. E., WALKER, S., HANCOCK-BEAULIEU, M., AND GATFORD, M. 1994. Okapi at TREC-3. In D. K. HARMAN Ed., *Proceedings of the Third Text Retrieval Conference (TREC-3)* (Gaithersburg MD, November 1994). U.S. National Institute of Standards and Technology. NIST special publication 500-225.

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.