ACSys TREC-7 Experiments*

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

January 27, 1999

Abstract

Experiments relating to TREC-7 Ad Hoc, HP and VLC tasks are described and results reported. Minor refinements of last year's Ad Hoc methods do not appear to have resulted in worthwile improvements in performance. However, larger benefits were gained from automatic feedback than last year and concept scoring was very beneficial in the Manual Ad Hoc category. In the Automatic Ad Hoc category title-only performance seems to have suffered more severely than long-topic from a number of lexical scanning shortcomings and from an excessive stopword list. The HP track was used to validate the usibility of the combination of PADRE and the Quokka GUI. In the VLC track, the 100 gigabyte collection was indexed in under eight hours and moderately effective queries were processed in less than two seconds.

1 Introduction

The work reported here comprises a number of text retrieval experiments conducted within the framework of TREC-7 and addressing questions of interest in the following research areas: Scalable information retrieval; Query term weighting; Concept-based relevance scoring; User-efficient retrieval interfaces and Automatic query generation.

ACSys completed Automatic and Manual Adhoc, High Precision and VLC tasks.

2 Method

2.1 Relevance Scoring Methods Employed

As in TREC-6 [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 (both measured in bytes).

On average, the Okapi probabilistic model performs well but there are cases where it does not. For example, a query derived from the topic "vitamins and health" is likely to comprise a set of vitamin words and a set of health words. For a document to be relevant, it should contain instances of words from both sets, but a sum-of-weights formula such as Okapi does not recognise this.

ACSys/ANU TREC submissions have focussed on scoring methods which tend to reward the conjunction of multiple concepts (e.g. the *vitamin* and *health* concepts in the example above). Our TREC-4 manual adhoc approach used *distance-based scoring* [Hawking and Thistlewaite 1995; Hawking and Thistlewaite 1996]. In TREC-5, ACSys/ANU manual queries were scored in the same way and a largely unsuccessful effort was made to provide semi-automatic assistance in query generation.

Neither TREC-4 nor TREC-5 manual runs involved any interaction, a fact which probably led to easily avoidable failures on some topics. As might be expected, distance-based queries perform more successfully in an interactive environment. [Cormack et al. 1997] In TREC-6, ANU/ACSys [Hawking et al. 1997] introduced a weaker method for rewarding concept co-occurrence *concept scoring* in the manual categories which was shown to produce a small but worthwhile benefit, on average.

This year, concept scoring was used in the title-only Automatic Adhoc submission, where each non-stop title word was assumed to represent a concept, and in Manual Adhoc, where concepts generated in the same way were modified by the human searcher. No means were to hand for automatically assigning terms from the description and narrative fields to concepts and, consequently, concept scoring was not used in the longer Automatic Adhoc categories. However, in these runs, a *Term Coordination* measure was used to reward documents with a wide spread of query terms. (See Section 2.1.3.)

2.1.1 Frequency Scoring

The basic Okapi relevance scoring method defined in Equation 1 will from now on be referred to as *frequency scoring* to distinguish it from the other methods.

2.1.2 Concept Scoring

As previously mentioned, 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 $s = (k_c c_1 + 1) \times \ldots \times (k_c c_n + 1)$. A value of $k_c = 1$ was used in concept scoring experiments reported here.

2.1.3 Term Coordination

When term coordination was in force, scores derived from the basic Okapi formula were multiplied by a term-coordination factor as follows:

$$S' = \frac{k_t + num\_qterms\_present}{k_t + num\_qterms}.S$$

A value of $k_t = 10.0$ was used in term coordination runs reported here.

2.2 Run-naming Convention

All ACSys TREC-7 runs consist of the string `acsys7` followed by a suffix which indicates the task category. Additional suffixes may be appended to differentiate runs in the same category. Table 1 lists the runids of official runs.

Table 1: Runids for the official ACSys submissions. Runids of submissions in the VLC track give the size of the collection, and the number of terms in the queries used. Thus `acsys7_100_2` is a VLC run over 100 gigabytes of data using 2-term queries.

Name	Category
`acsys7as`	Automatic Ad Hoc, short topic (T)
`acsys7am`	Automatic Ad Hoc, medium topic (T+D)
`acsys7al`	Automatic Ad Hoc, long topic (T+D+N)
`acsys7mi`	Manual Ad Hoc, interactive
`acsys7hp`	High Precision Track
`acsys7_`g`_`n	VLC Track, n-term query over g gigabytes of data.

2.3 Training

Training was carried out using the TREC-7 data and TREC-6 topics after removing the Congressional Record documents from the TREC-6 qrels.

2.4 Hardware and Software Employed

The PADRE retrieval system used in previous TRECs has undergone further evolution, mainly to improve efficiency. The current version is known as PADRE98 and it was used in all experiments reported here. A Dell Latitude laptop PC running Linux and a Sun Ultra-1 workstation were used in the Adhoc runs. In the VLC track, a cluster of eight DEC Alphas was employed.

Interactive query modification was carried out using the `quokka` graphical user interface to PADRE.

2.5 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 $\alpha = 0.05$.

2.6 Automatic Query Generation

The basic approach to automatic query generation was as described in last year's TREC paper, except for the changes described in this section. [Hawking et al. 1997]

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

1. How can the Concept scoring method be used with automatically generated queries?
2. Can the use of frequency within the topic as a query-term weight be improved upon?

The basic relevance scoring method used by ACSys (see Equation 1), makes use of a query term weight q_t. In TREC-6, q_t was simply a count of how many times the query term occurred in the part of the topic statement being used.

In TREC-7, an attempt was made to calculate better query term weights by using a modified Okapi formula, in which the accessible part of the query statement was treated as a document and in which *df* values were derived from the body of text represented by TREC topics 1-400. The main effect of this is to reduce the weight of terms occurring in many topic statements (such as "document", "identify", "relevant", etc.) and to reduce the boost given to terms occurring many times in the topic statement.

A simplistic attempt was also made to identify terms which were more central to the topic, by artificially boosting the frequency of occurrence for non-stopwords in the title field by a constant (k_e). A value of $k_e = 1.5$ was found to be effective.

As previously mentioned, in the title-only run it was assumed that each non-stop title word represented a separate concept. In addition, phrases were not generated for the title-only task, as training suggested that they harmed performance in this category (only).

2.7 Relevance Feedback

The pseudo-relevance feedback mechanism based on hotspots (passages) in the retrieved documents was used almost unchanged from TREC-6. However, the changed method for calculating query-term weights necessitated a change in the interpretation of the w_0 parameter (the query-term weight of the best term selected by feedback). In TREC-6, it was a constant ($w_0 = 0.75$) but in TREC-7 it was interpreted as a fraction to be multiplied by the maximum query-term weight of any of the original terms.

2.8 Parameters Used in Official TREC-7 Runs

Table 2: Parameter settings used in the official Automatic Adhoc runs. k_e - emphasis added to terms occurring in the title field; $avlen_q$ - value used for average topic length in calculating query-term weights; k_t - term coordination constant; k_c - concept scoring constant; T - Number of top-ranked documents examined during relevance feedback (RF); p - Proximity range (in characters) used to define RF hotspots; n - Number of RF terms extracted; w_0 - multiplied by maximum query-term weight to give weight of best RF term. Times per query are elapsed times in seconds measured on a 167 MHz,256 Mbyte Sun Ultra-1. Figures in parentheses are the corresponding figures for a 266 MHz Pentium-2 Dell Latitude laptop.

	`acsys7as`	`acsys7am`	`acsys7al`
Topic fields	T	T + D	T + D + N
k_e	0	1.5	1.5
$avlen_q$	20	1	15
k_t	NA	10.0	10.0
k_c	1.0	NA	NA
T	20	20	20
p	500	500	500
n	20	20	20
w_0	0.4	0.3	0.2
#terms	2.4	6.2	17.0
#phrases	0	3.8	10.5
Time/query	10.9(17.0)	16.5(23.1)	33.3(41.9)

Parameters used in Automatic Adhoc runs are detailed in Table 2.

2.9 Query Optimisation

In various TREC tasks, such as HP and VLC, query processing was optimised by the simple expedient of ranking the query terms in order of decreasing q_t and processing only the top k of them. This behaviour is controlled by including `MAXTERMS = ` k directives in the query stream.

3 Results

3.1 Automatic Adhoc Results

Table 3: Average Precision performance of ANU/ACSys Automatic Adhoc runs relative to 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 (in both cases relative to *all* automatic runs, not just those using the same topic fields) are tabulated in the rightmost two columns. The rank is relative to automatic runs of the same topic length. The starred run was an unofficial run performed under official conditions and prior to the deadline. There were 50 topics.

Run-id	Category	Mean	Rank	#best	#≥med.
acsys7as	T	.2045	8	2	26
acsys7am*	T+D	.2230	6*		
acsys7al	T+D+N	.2659	7	2	42

Table 4: The same runs as in table 3, compared on the basis of overall recall (percentage of known relevant documents retrieved, averaged across 50 topics).

Run-id	Category	Percent	#best	#≥med.
acsys7as	T	61.7	2	37
acsys7am*	T+D	68.1	6*	40
acsys7al	T+D+N	72.7	4	44

Table 5: The same runs as in table 3, compared on the basis of P@20. Best and median data was not available.

Run-id	Category	Mean
acsys7as	T	0.337
acsys7am*	T+D	0.359
acsys7al	T+D+N	0.439

Results for Automatic Adhoc runs are summarised in Tables 3, 4 and 5. The long topic run outperformed the title-only run by considerable margins (30%, 30%, and 18% for average precision, P@20 and recall respectively). The gaps between long topic and title-plus-description are smaller (19%, 22% and 7% respectively) but still statistically significant. In training runs (see Section 2.3), these differences had been much smaller.

3.2 Automatic Adhoc Discussion and Conclusions

This year's attempted improvements in automatic query generation appear to have been founded on too little training data. Relative to comparable runs performed using last year's methods, this year' runs achieved "gains" of +6%, -1% and -3% in average precision for long, medium and short topics. None of these differences were statistically significant. This was disappointing, as training had been particularly focussed on the shorter length topics.

With the assistance of the Okapi group, an analysis was undertaken of why ACSys short-topic queries were relatively poor. The following reasons are believed to explain the major part of the differences:

1. Lexical and stop-word issues had large effects on certain topics. In the topic *R&D drug prices*, R&D was eliminated in lexical scanning and prices was (for some peculiar reason) on the stop list. The resulting query was the ineffective *drug*. The version of PADRE in use treated only the first 12 characters of a word as significant, but the same restriction did not apply to the query generator. Consequently, results for *oceanographic vessels* (including a 13-letter word) were quite poor.

2. Okapi made use of a small number of pre-defined synonym classes. On certain topics, the addition of synonyms (such as *Malvinas* to *Falkland* and *channel tunnel* to *chunnel*, caused a considerable gain.

3. Passage retrieval. The Okapi group reported a small overall gain from use of passages prior to pseudo relevance feedback. ACSys did not use passage retrieval.

Without removal of these differences, it is not possible to compare the relative effectiveness of the two variants of the BM25 formula used, nor to compare the results of the relevance feedback methods.

4 Manual Query Generation

Manual AdHoc, Official Run `acsys7mi`**, Corrected Runs** `acsys7mi2` **and** `acsys7mi2rf`

4.1 Manual Query Generation Process

A reasonably experienced user (the first author) interactively generated a set of manual queries starting with an initial automatic set. The initial queries were generated from the full topic descriptions and were similar to the queries used in `acsys7al` but used neither feedback nor phrases.

Concepts corresponding to the title words were generated automatically, with non-title words being initially assigned to the last concept. The Quokka GUI allowed user-efficient deletion of terms and assignment of terms to concepts.

The task was approached in a similar fashion to the High Precision track except that a) more time was allowed and b) the goal was to produce a generally-useful single query which could then be run to achieve good results, particularly on the early precision measure.

Query modification proceeded in a number of phases:

1. Blind (no reference to documents) refinement of the query by: a) restructuring (if necessary) the concepts, b) removing terms thought to be of low value, c) assigning terms to the appropriate concepts, and d) adding new terms thought to be useful. The goal was to produce an initial document ranking with as many relevant documents as possible in the first ten.

2. The query was then run by PADRE with a restriction on the number of active document accumulators and light query optimisation (MAXTERMS=15). These steps were intended to reduce user waiting time. The titles of the top 100 documents were displayed in a scrolling window, each with coloured squares indicating the presence of concepts within the document.

3. The user then examined documents to evaluate the performance of the query. Sometimes they were chosen from the top of the ranking; at other times documents flagged as containing evidence for all of the concepts were chosed from further down the list. Query terms occurring within displayed documents were automatically highlighted in the colour assigned to the concept for which they provided evidence. A decision was made as to the document's relevance and recorded as a red or green square beside the document title.

4. After reading a few documents, it was usually clear that one of the following applied:

   (a) The query was performing very well and needed no further modification.

   (b) The query was performing well, but a list of additional terms had been identified which could improve it. These were added and the query re-run.

   (c) Irrelevant documents were being returned due to distractor (eg. ambiguous) terms. In this case, terms might be removed (sometimes with addition of new terms) or negative-weight terms added and the query re-run.

   (d) The query was unbalanced and gave too much weight to some concepts. An indication of this appears in the pattern of coloured concept squares in the document list. This was addressed by some combination of: adjusting the query-term weights to increase the emphasis on neglected concepts; and adding new terms to neglected concepts.

5. Steps 2-4 were repeated until the user was satisfied with the proportion of relevant documents in the first ten or so documents or gave up.

6. The last-run version of the query and the top 1000 ranking resulting from running it were saved for submission.

The median time taken per topic was 10.6 minutes. A total of 533 minutes was required for all 50 topics. After completion of the task, any unjudged douments within the top 10 for each ranking were judged (time not recorded) in order to self-assess the performance of the queries. On average, 7.6 documents in the first ten were self-judged to be *appropriate*, which was felt to be a good result. Again based on self-assessment, there were only six topics for which less than half of the top 10 were appropriate. The definition of *appropriate* was weaker than the definition of *relevant*. Appropriate documents basically matched the query but not necessarily the topic definition of relevant. They may not, for example, have provided the specific examples or specific information demanded by the topic but not by the resulting query.

It was observed that long documents occurred very rarely among the first ten retrieved. Almost none of the documents examined were longer than two or three screenfuls.

4.2 Manual Adhoc Results

Table 6 records the results of various manual runs with two long-topic runs for comparison.

Contrary to previous experience, the results for the frequency-scored version of the official manual run (`acsys7mif`) seem to show that concept scoring actually caused harm. Subsequent investigation revealed that this was a case of death by misadventure! It transpired that MAXTERMS=15 directives had been inadvertently left in the official run. Removal of the offending directives showed that this mistake had caused a 20% drop in average precision, a 7% drop in P@20 and a 9% drop in recall. All these differences were statistically significant, but may be underestimates because the effect of the MAXTERMS directive during the interactive phase is likely to have been more deleterious than intended or expected.

The harm caused by the term limit to the frequency-scored version of the official run was minimal because, in that case, only low-value terms were not processed. A frequency scored version of the corrected manual run showed that, in fact, concept scoring achieved a gain of 13% (significant) on average precison, a gain of 11% on P@20 at the expense of an apparent loss of 4% on recall (not significant).

Table 6: Results for Manual runs. Averages across 50 topics. Data for the automatic queries from which the manual queries were derived is also included. As mentioned in the text, the official `acsys7mi` run inadvertently included an optimisation directive which harmed performance. The corrected run `acsys7mi2` is identical but for the removal of this directive. Other runs included here show the benefit of concept scoring and pseudo relevance feedback. The suffix `nf` appended to a runid indicates that that the new run was identical except that relevance feedback was not used. Similarly, `rf` indicates a run variant in which relevance feedback was used and `f` indicates a variant in which frequency scoring was used instead of concept scoring.

Run-id	Description	Ave. Prec.	P@20	Recall
`acsys7alnf`	`acsys7al` minus rel. feedback	.2260	.400	69.3%
`acsys7mi`	Official manual ad hoc	.2669	.486	63.0%
`acsys7mif*`	`acsys7mi` minus concept scoring	.2786	.472	71.5%
`acsys7mi2`	Corrected `acsys7mi`	.3196	.525	69.0%
`acsys7mi2f*`	`acsys7mi2` minus concept scoring	.2816	.473	71.8%
`acsys7mi2rf`	`acsys7mi2` with rel. feedback	.3401	.530	74.2%

4.3 Manual Adhoc Discussion and Conclusions

The effect of the manual intervention can be gauged by comparing the corrected manual run with the `acsys7al` run a) without feedback, and b) with it. In the no feedback case, the former performed 41% better on average precision (significant) and 31% better on P@20 (significant). Recall levels were effectively identical. In the feedback case, the manual run performed 28% better on average precision (significant) and 21% better on P@20 (significant). Recall levels were again effectively identical.

In the non-feedback case, query processing times were substantially reduced in the (corrected) manual run, from an average of 17.3 sec. to 4.55 sec.

From the results of the corrected manual run, it is clear that a relatively small amount of manual editing can create quite dramatic improvements in precision and query processing speed.

Concept scoring again produced a worthwile gain in performance.

4.4 Benefits of Automatic Feedback

Table 7: The effect of automatic feedback on various query sets. Scores shown for the various runs apply to the no-feedback case. The percentage gain due to feedback is given in parentheses. Asterisks indicate statistical significance.

Run-id	Ave. Prec.	P@20	Recall
`acsys7as`	.1677(+22%*)	.317(+6%)	54.1%(+14%*)
`acsys7am`	.1940(+15%*)	.344(+4%)	61.8 %(+10%*)
`acsys7al`	.2260(+18%*)	.400(+10%*)	69.3%(+5%)
`acsys7mi2`	.3196(+6%*)	.525(+1%)	69.0%(+7%*)

Table 7 shows that relevance feedback has worked consistently well for all three automatically generated query sets and for the (corrected) manually modified set. All differences in average precision were statistically significant. Three of the four query sets also showed significant gains in recall. It is interesting that the only query set which did not show a significant change in recall was also the only one to show a significant improvement in P@20.

5 High Speed (... sorry, Precision) Track

The High Precision task is probably the best overall test of a retrieval system's performance (provided that comparisons are not confounded by user variation.) Good performance on the HP task requires acceptable speed and effectiveness of the retrieval system and also an effective user interface but may be unaffected by small advantages on any single dimension. The HP task is also a good framework in which to evaluate the benefit or otherwise of bells and whistles. If users can't take advantage of them on the HP task, how practically useful are bells and whistles?

For this year's HP track, PADRE and the Quokka were used unaltered, except for the addition of a timer to the Quokka. All 50 topics were done in sequence by one user over the course of a single day. To ensure all judgements were entered within the five minutes, a time stamp was entered with each judgement according to the Quokka's timer. The timer was started immediately a new topic was visited, and the final sumbission was generated by taking all judgements with a timestamp of less than 300 seconds.

No attempt was made to overlap query processing with human reading of documents.

PADRE was run several times per query, probably two or three times on average. Three or four word queries were usually used on the first run, then four or five extra terms would be added over the course of the subsequent runs. The colour coded highlighting of query terms in the Quokka was useful for making fast relevance judgements on viewed documents.

Subsequent work on the HP task is likely to focus on analysing the amount of time spent by the user on the various activities such as: a) composing, editing and typing queries; b) waiting for PADRE to process queries; c) waiting for Quokka to display selected documents; d) reading relevant documents; e) reading irrelevant documents. Such an analysis should identify the most profitable areas to speed up, and how to set the balance between speed and effectiveness.

Our impression is that further improvement to PADRE query response time and reduction in the time taken to load a document in the Quokka are likely to be most beneficial. Adding new query expansion mechanisms such as automatic or interactive relevance feedback could both improve the quality of results and the amount of useful highlighting in each document. However, PADRE feedback is currently very slow and would need to be accelerated significantly for these mechanisms to deliver a nett benefit.

6 VLC Track

Although ineligible to win an ACSys medal, it was nonetheless our goal to meet the criteria. This constituted a considerable challenge, as PADRE's speed in last year's track was far lower than the required level.

6.1 Hardware Employed

Experiments were conducted using eight 266 MHz, 128MB EV5 DEC Alpha workstations connected by a 10 Mb/sec Ethernet network. An elderly SPARC 10 (60 MHz, 96MB) connected by a shared 10 Mb/sec network was used as the user interface. This was essentially the same hardware configuration as last year, but this time no use was made either of the RAID box connected to one of the workstations or of the faster ATM network connecting the machines. However, this time, a 9 gbyte external SCSI disk was connected to each node. The small internal disk on each node was used by the operating system and for swap space.

6.2 Indexing

As in the past, the data was divided approximately evenly across the available workstations. If the data allocated to a workstation was much more than one gigabyte, the data was divided into separately indexed *chunks*.

The indexing process for a chunk involves building a compressed inverted file, then post-processing it to produce an uncompressed but reasonably compact inverted file in which each posting is a (docno, score) pair. The score is a quantised version of the Okapi weight taking into account df, tf and document length.

During indexing (and during query processing), the various dictionary, document and index files are mapped into virtual memory using the `mmap()` call and treated as arrays. This reduces system overhead, and avoids both the need to manage buffers and the need for explicit I/O calls.

A reasonably efficient algorithm has been implemented to merge several separate chunk indexes into a single index.

6.3 Indexing Results

Last year, PADRE required 15.6 hours to index the 20 gB VLC. The advent of local disks on each of the workstations allowed the process to be parallelised and this year it took only 7.38 hours to index 100 gbytes of data. With better load balancing, this figure would have been 5.7 hours. These figures include the time to pre-compute relevance score contributions for every possible term-document combination.

Table 8 gives the indexing times for BASE1, BASE10 and VLC2 collections using all eight workstations. The time for BASE10 appears anomalous. This could be because of poor load balance caused chunks which were excessively large on one or more nodes, resulting in excessive virtual memory system activity.

Indexing of BASE1 was also carried out on one workstation. The time taken was 0.800 hours, a factor of 18.4 longer than in the eight workstation case. This is very likely to be due to virtual memory effects.

6.4 Query Processing Speed Results

Last year, PADRE required an average of 50.6 seconds to process queries over the 20 gbyte VLC. With more heavily optimised queries and considerable tuning, it was found possible to reduce average query processing times over 100 gbytes of data to 2.74 seconds, using the indexes and hardware described above.

Further improvement was hampered by the fact that the original data was divided into 12 (in one case 13) separately-indexed chunks per workstation. Consequently, there were 12 (in one case 13) separate term dictionaries and indexes on each machine. This organisation resulted in large numbers of page faults and consequent delay. Only a very small proportion of the elapsed query processing time is CPU time.

To address the problem, separately created data structures were merged. Groups of three (in one case four) data structure sets were merged, reducing the number of components per workstation to four. Merging was done in parallel across the nodes and the longest elapsed time was 0.35 hours, bringing total indexing time to **7.73 hours**. Having merged in this way, average query processing time dropped to 0.887 seconds for 2-term queries. This is a factor of about 280 better than last year, taking into account the data scale-up, on similar, even less-expensive, hardware. Average query processing time rose to 1.47 seconds for 5-term queries.

It appears that presentation of rankings (mostly looking up the docid) are taking about 0.3 sec per query and there is probably at least 0.1 sec of fixed overhead. It is therefore expected that completing the merging process to the point where each node has but one set of structures might result in reducing average query times down to about 0.5 sec but no further with present hardware. On the other hand, increasing the degree of merging still further is expected to allow the data size per node to be increased substantially while keeping query processing time below 1 second.

Additional memory or parallel disk I/O on each workstation could bring considerable benefit to both indexing and to query processing.

6.5 Speed Ups

There are a number of reasons why this year's processing speed is better:

1. Each workstation has its own local disk,
2. Queries are more heavily optimised,
3. Using a chair (eg. a block index) in front of the term table (to improve locality of reference and reduce page faults),
4. Using reference stats instead of communication between nodes to obtain dfs,
5. Relevance scores are pre-computed during indexing (in other words the inverted file contains the actual score for every term-document combination),
6. Limiting the number of document accumulators,
7. Using radix sorting of results (less than two full passes through the document accumulators are needed),
8. Increasing the size of data represented in each separate index.

6.6 VLC Effectiveness Results

Unfortunately, the dramatic improvements in speed have been achieved at the cost of retrieval effectiveness. Precision at 20 documents retrieved for the two-term (0.298, ranked 18/18) and five-term runs (0.442, ranked equal 12/18) over 100 gigabytes was at the bottom end of the table of participating groups. The median P@20 score for the 18 runs was 0.525.

However, when the six runs corresponding to manually generated or automatic full-topic queries are eliminated, the median average precision over 100 gigabytes drops to 0.442, meaning that, technically, the ACSys 5-term run did in fact (though barely) satisfy the medal conditions.

6.7 VLC Scalability Results

Run `acsys7_1_5` was repeated using only a single Alpha (run `acsys7_1_5S`) instead of eight. As expected, P@20 results were very similar (0.141 v. 0.139). The small difference (involving a total of only two relevant documents) is believed to be due to arbitrary differences in ranking of equal-scoring documents. The average query processing times were quite similar (0.061 sec. and 0.086 sec. respectively) indicating that parallelism introduces almost no query processing benefit in the BASE1 case in the conditions of the experiment.

Note that in the 1 gB, eight-node case, there is sufficient RAM (128MB per node) for all necessary data structures to remain resident, dramatically increasing the chance that document numbers will be memory resident.

The scale-up between the five-term query runs is shown in the three middle columns of Table 8. As may be seen, there is a dramatic increase in P@20 from 1 gB to 10 gB and a less dramatic but still significant jump from 10 gB to 100 gB. In the case of query processing time, the scale-up from 10 gB to 100gB is almost the same as the scale-up in collection size whereas the time scale-up from 1 gB to 10 gB is much less than the scale-up in collection size. It is almost certainly the case that the data-independent costs in query processing (query parsing and broadcast, document identifier look-up, result merging and

Table 8: Performance of PADRE98 on the three different collection sizes, averaged over the 50 Ad Hoc topics. Except for the results in the right-most column, all data relates to the same set of 5-term queries. All runs used eight DEC Alpha workstations. Query processing times are in elapsed seconds. Each is the average of three runs and each group of three runs was preceded by a warm-up query (a 51-term manual query for topic 254). Indexing time is in elapsed hours and is the time taken by the slowest node). Indexing was performed only once. Figures in parentheses give the scale-up factor over either BASE1 or both BASE1 and BASE10 as appropriate.

Measure	BASE1	BASE10	VLC2	VLC2 (2-term queries)
Runid	`acsys7_1_5`	`acsys7_10_5`	`acsys7_100_5`	`acsys7_100_2`
P@20	0.139	0.321 (2.31)	0.442 (3.18, 1.38)	0.298 (2.14, 0.928)
Indexing time	0.0434	1.71(39.4)	7.73(178,4.52)	-
Time/query	0.061	0.168 (2.75)	1.468 (24.1, 8.74)	0.887 (14.5, 5.28)

presentation) are responsible for the bulk of the time spent in the 1 gigabyte case. If the fixed overheads are assumed to be about 0.05 seconds per query, then the ratios correspond much more closely to the data scale-up factor.

The two-term run over the full collection was made: a) to achieve sub-second query processing times, and b) to investigate whether the expected increase in P@20 due to collection scale-up could be traded for a lower query processing time scale-up factor. As may be seen in Table 8, the two-term queries achieve more than twice the P@20 over 100 gB as do the five-term queries over 1 gB and almost the same P@20 as do the five-term queries over 10 gB, while achieving a significant saving in processing time.

6.8 VLC Discussion and Conclusions

At this stage, it is unclear why precision results were as relatively bad as they were. Possibilities include:

1. Inaccurate *df* estimates derived from last year's VLC track baseline.
2. Errors due to quantisation of pre-computed scores.
3. Too-draconian limit on number of document accumulators with possible bias against particular collections.
4. Query optimisation too aggressive or terms badly ranked.
5. Thresholding of postings too aggressive or completely ill-advised.
6. It is possible (but unlikely given the similarity to methods used by the Okapi group) that PADRE98's basic retrieval methods do not work as well on Web data.
7. Bugs. (Surely not!)

The combination of techniques such as radix sorting, relevance score pre-computation (with uncompressed index files) has the effect of reducing the CPU cost of query processing. At the same time, the combination of index thresholding and the MAXTERMS style of query optimisation ensures that postings lists are kept short and therefore limits the amount of data to be transferred from disk. The result is that disk seek time becomes the dominant factor. In PADRE98, the processing of each query term over each chunk of text is likely to require one I/O request (with seek and rotational latency) to locate the appropriate entry in the term dictionary and another to retrieve the postings list. In addition, a disk I/O is nearly always needed to obtain the name of a ranked document.

7 Conclusions

1. The small innovations introduced in the Automatic AdHoc runs do not appear to have been beneficial on the TREC-7 task. However, the T+D+N run again performed at a quite respectable level. In last year's Automatic Adhoc tasks, the relevance scoring model used by ACSys worked much better for the T+D+N task than for the shorter topics. Attempts to improve relative performance on shorter topics in TREC-7 seem to have been unsuccessful.

2. Once again, we have observed that results obtained in training using only one set of 50 topics do not necessarily generalise. When time permits it is always worth training on multiple topic sets.

3. ACSys Manual AdHoc participation seems to indicate that a small (highly topic-dependent, but averaging 10 minutes) amount of human intervention can significantly improve on automatically generated queries as far as P@20 and average precision are concerned. Automatic query generation may have "hit the wall" but humans are on the other side of it!

4. Participation in Manual AdHoc and High-Precision tasks has confirmed the view of the users (also authors) that the PADRE/quokka combination is quite usable and reasonably effective. Comparison with other systems on these tasks does not seem to be terribly fruitful given that there is no control over individual (human) differences. Future work is likely to focus on identifying sources of delay or inefficiency which might be eliminated.

5. The speed-up and improvement in "bang-per-buck" of the ACSys VLC runs compared to TREC-6 was very pleasing. It remains to be seen whether the loss of effectiveness relative to last year can be reversed without sacrificing the speed gains. It was pleasing to achieve our VLC goal, particularly when, at times, it seemed so far away.

Acknowledgements

We are indebted to Steve Walker of the Okapi group for his cooperation in providing queries, additional results and information about methods. This help was invaluable in analysing the performance of the ACSys runs.

Bibliography

Cormack, G., Palmer, C., To, S., and Clarke, C. 1997. Passage-based refinement: Multitext experiments for TREC-6. In E. M. Voorhees and D. K. Harman Eds., *Proceedings of the Sixth Text Retrieval Conference (TREC-6)* (Gaithersburg MD, November 1997), pp. 303–320. U.S. National Institute of Standards and Technology. NIST special publication 500-240.

Hawking, D. and Thistlewaite, P. 1995. Proximity operators - so near and yet so far. In D. K. Harman Ed., *Proceedings of the Fourth Text Retrieval Conference (TREC-4)* (Gaithersburg MD, November 1995), pp. 131–143. U.S. National Institute of Standards and Technology. NIST special publication 500-236.

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 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.

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.

VOORHEES, E. M. AND HARMAN, D. K. Eds. 1997. *Proceedings of the Sixth Text Retrieval Conference (TREC-6)* (Gaithersburg MD, November 1997). U.S. National Institute of Standards and Technology. NIST special publication 500-240.