Proximity Operators - So Near And Yet So Far

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

December 6, 1995

Abstract

Testing of the hypothesis that good precision-recall performance can be based entirely on proximity relationships is a focus of current TREC work at ANU. PADRE's "Z-mode" method (based on proximity spans) of scoring relevance has been shown to produce reasonable results for hand-crafted queries in the Adhoc section. It is capable of producing equally good results in database merging and routing contexts due to its independence from collection statistics. Further work is needed to determine whether Z-mode is capable of achieving top-flight results. A new approach to automatic query generation designed to work with the shorter TREC-4 queries produced reasonable results relative to other groups but fell short of expectations based on training performance. Investigation of causes is still under way.

1 Introduction

The Parallel Document Retrieval Engine (PADRE) has previously demonstrated that full text scanning methods supported by parallel hardware permit powerful query constructors and rapid response to changing document collections [5, 6]. Since then, the addition of parallel-disk-resident inverted file indexes and dictionaries has potentially extended data handling capacity to the terabyte level, with an expectation of reasonable response times but some loss of flexibility [7]. Recent work [8] has added an `http` interface and enabled multiple simultaneous user sessions.

PADRE derives from an earlier ANU parallel system called PADDY [3, 12] which included a partial emulation of the functionality of the PAT text search program developed at the University of Waterloo [1, 2].

From an Information Retrieval perspective, PADRE now allows greater choice of relevance estimation functions (including one based on proximity spans [1]), is capable of simulating the operation of boolean style queries and provides a means of restricting the effect of precision-enhancing operations to only those documents which pass a test of basic relevance.

TREC runs based on the new improved PADRE were submitted in the Automatic Adhoc, Manual Adhoc and Database Merging categories. As was the case last year, official runs did not make use of inverted files but instead used full-text scanning methods over memory-resident data. A 512-node Fujitsu AP1000 (8 gigabytes of RAM) in Kawasaki was used for the Manual Adhoc submission. ANU's 128-node AP1000 (2 gigabytes of RAM) was used for most training

[1] Very similar to the one described in the University of Waterloo paper in these proceedings. Despite the fuzzy ancestral relationship described above, the two relevance models were developed completely independently. In fact the version of PAT with which we are familiar includes no concept of relevance at all.

runs and for the Automatic Adhoc and Database Merging runs. Previous training runs having suggested that merging results from CD2 with separately obtained results from CD3 did not introduce significant distortion, this approach was used for training and in the Automatic Adhoc submission.

The development at ANU of an experimental parallel file system [11], capable of loading a gigabyte of data in 20 seconds, has dramatically increased the number and scope of experiments which can be conducted in a given time.

The present paper describes the new PADRE features which underpin our submissions, documents the query generation processes and reports and analyses the results obtained. In addition, some further related experiments are reported.

2 Recent Extensions To PADRE Query Language

Since TREC3, a number of extensions and refinements to the query engine have been implemented.

2.1 Choice Of Term-Weighting Schemes

The query writer now has considerable freedom to select variants of the well-known *tf.idf* weighting scheme.

A document's estimated relevance R_d is still computed according to the following formula:

$$R_d = \sum_{t=1}^{k} (I_t \times r_{(t,d)}) \tag{1}$$

where:

k is the number of terms,

I_t is the manually assigned importance of term t, and

$r_{(t,d)}$ is the relevance of a document d due to term t.

but there are now a number of choices for the the calculation of $r_{(t,d)}$, such as:

$$r_{(t,d)} = \frac{f_{(t,d)}}{\sqrt{F_t \times l_d}} \tag{2}$$

$$r_{(t,d)} = \frac{f_{(t,d)}}{\log{(F_t \times l_d)}} \tag{3}$$

$$r_{(t,d)} = \frac{f_{(t,d)}}{\log F_t \times \log l_d} \tag{4}$$

$$r_{(t,d)} = \frac{f_{(t,d)}}{\log l_d} \tag{5}$$

$$r_{(t,d)} = 1.0 \; if \;\; t \; is \; present \; in \; d, \;\; otherwise \; 0.0 \tag{6}$$

where:

$f_{(t,d)}$ is the frequency of term t in document d,

F_t is the frequency of term t in the entire collection, and

l_d is the length of document d

F_t may optionally be defined as the number of documents containing term t rather than the total number of term occurrences.

2.2 Mandatory Inclusion Or Exclusion of Documents

PADRE now records *include* and *exclude* flag bits for each document in addition to accumulated positive relevance, accumulated negative relevance and maximum individual contribution to positive relevance. Accumulated document scores may be reset at any time without affecting the flag bit settings.

Commands are provided to set the flags for each element of the document set (or its complement) containing one or more members or the current match set. Documents may also be excluded from the relevant set on the basis that positive evidence is too weak, negative evidence is too strong or that positive evidence comes too heavily from a single source. Similarly, documents may be mandatorily included on the grounds that negative evidence is weak or that positive evidence is strong.

The exclude bit can be used to implement a type of query in which recall-oriented terms are used to define a *universe of discourse* set of documents prior to searches for precision-enhancing terms. Documents outside the universe of discourse remain excluded no matter how many occurrences of the latter terms they may contain. As an example, the universe of discourse may be defined as those documents containing at least m from a basic set of n key terms (or all of them). The precision phase searches for terms which do not by themselves imply relevance but whose presence in a document passing a general test of relevance increases the probability that it is actually relevant. For example in topic 204, the recall phase could identify documents dealing with *nuclear power plants* and the precision phase could boost the scores of documents in this universe of discourse which contain references to US-related terms and terms such as *gigaWatt*. Occurrence of such terms outside the universe of discourse is unlikely to be significant.

2.3 PRELATE - A GUI For Query Generation

Last year's Manual Adhoc submission was plagued by structural and other errors in the queries. This year, a prototype graphical user interface was built using TCL/Tk in an attempt to avoid this problem by making very evident the structure of complex queries and by rendering certain types of error impossible and detecting and warning of others. This interface is known as PRELATE (the Padre REtrieval LAnguage Topic Editor).

In general, PADRE queries are hierarchical, as may be seen in the sample PRELATE screen dumps below.

PRELATE proved to be quite effective in composing complex manual queries. It was useful but not infallible in avoiding errors. It is however, not presently designed to support interactive queries and would need significant modification for effective use in this role. A future line of development may convert the tool into an *applet* for use with a TCL/Tk web browser.

3 Manual Query Generation

3.1 Philosophy

Last year's manual PADRE queries were constructed with heavy reliance on PADRE proximity relationships. However, key individual terms with low manually assigned weights were sometimes used to improve recall, out of fear that the proximity relationships might be too seldom satisfied. Document relevance was calculated by summing $tf.idf$ style weights derived both from singleton terms and from combinations of terms in proximity. This year, it was decided to pursue the proximity theme to its logical conclusion by basing relevance scores entirely on proximities.

Proximity relationships between terms intuitively seem to offer the potential to reduce spurious hits. To illustrate the point, an article was posted to the network news two years ago (and retrieved by an essentially boolean query) which contained widely separated occurences of the terms `Hawking`, `text` and `retrieval`. Its content bore no connection with one of the present authors and none with the field of IR. Had a proximity requirement been enforced, the document would not have been retrieved.

We adopted the following working hypothesis:

> *The closer together a set of intersecting terms, the more likely they are to indicate relevance.*

Proximity relationships can help to disambiguate different senses of the same word without the expense of semantic analysis. The word *bank* occurring in close proximity to *river, water, bridge* etc. is considered less likely to mean a financial institution than it would in other contexts.

Reliance on proximity has some of the flavour of passage-level approaches, but is cheaper to implement as it involves no semantic analysis of the text.

This year's Manual Adhoc submission was constructed entirely on the basis of *concept*s in proximity. No singleton term counted anything directly toward the relevance of documents. The presence of a concept in a document was signalled by a match against one of a [frequently large] set of alternative terms used to define it. These term sets were expanded considerably in order to improve recall. A proximity relationship between concepts will be referred to as a *concept intersection.*

3.1.1 Illustration of *Concept Intersections*

Let us use topic 203 ("What is the economic impact of recycling tires?"), to explain by example the process of generating queries based on concept intersections.

First, the topic was examined and three concepts were identified:

1. economic impact
2. recycling
3. tires

Words, phrases and regular expressions connoting each of the concepts were then generated from the query-writer's head, trying to take into account all the linguistic forms which might have been used to express the concept, including alternative spellings, plurals, different parts of speech and even mis-spellings.

As figure 1 shows, each set of terms connoting a concept were combined in an `anyof` operation. In the case of the *economic impact* concept, the match set arising from the regular

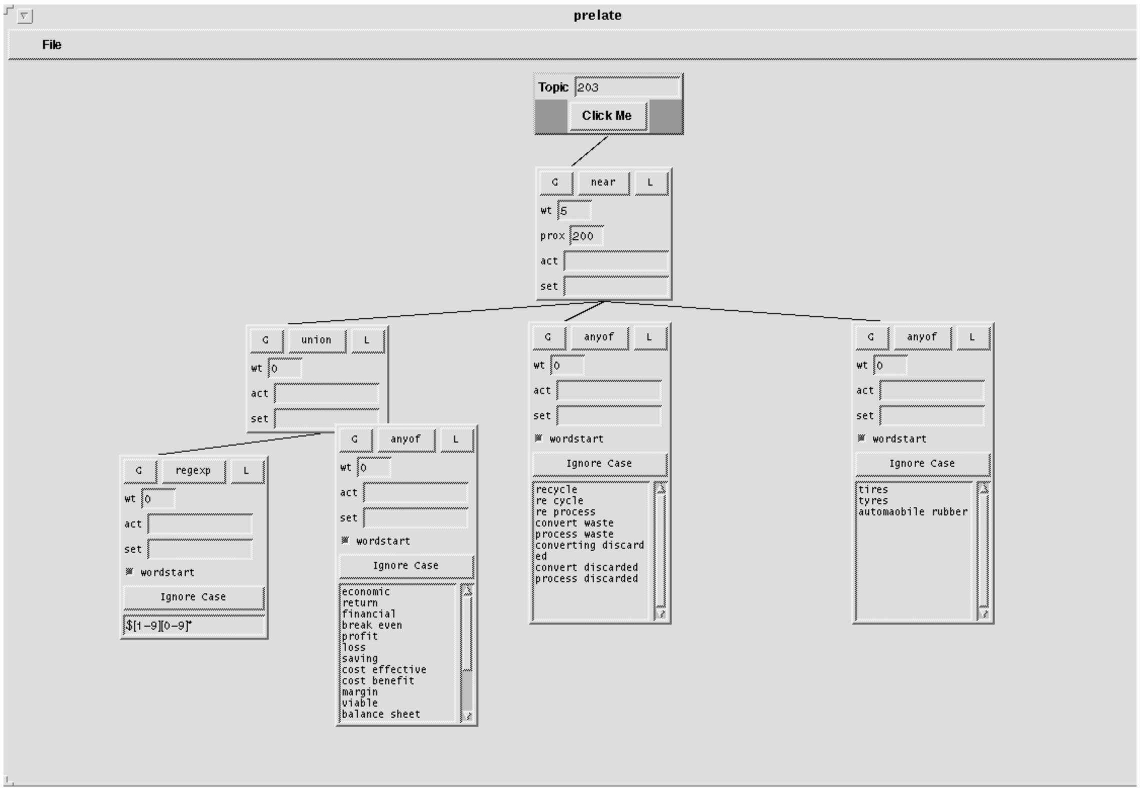

Figure 1: Screen dump of PRELATE representation of PADRE query for topic 203 relating to the economic impact of recycling tires.

expression searching for dollar amounts is combined with the results of the `anyof` applied to the other simple terms using a set union operation.

The presence of one of the concepts by itself is not strongly indicative of relevance. Even a document containing many occurrences of many terms from only one of the concepts is not much more likely to be relevant. For example, a document containing many references to balance sheets, dollar amounts, profits etc. but nothing at all relating to tires or recycling is unlikely to be relevant but would score highly in a uniform simple-term weighting scheme.

We postulated that the co-occurrence of all of the concepts within close proximity (a concept intersection) would be highly suggestive of relevance. Figure 1 shows that the concept intersection for query 203 is implemented using a `near` operator with a proximity of 200 characters and a weight of 5. All other operations have a weight of zero.

3.1.2 More Complex Proximity Relationships

Some topics gave rise to query structures involving multiple proximity relationships. This occurred in the following circumstances:

1. Cases where proximity relationships with small range were used to find "loose phrases" within a concept.

2. Cases in which a number of distinct proximity relationships independently imply relevance. That is, the presence of any one of the relationships $near(C_1, ..C_k)$, $near(C_{k+1}, ..C_l)$, ... suggests relevance.

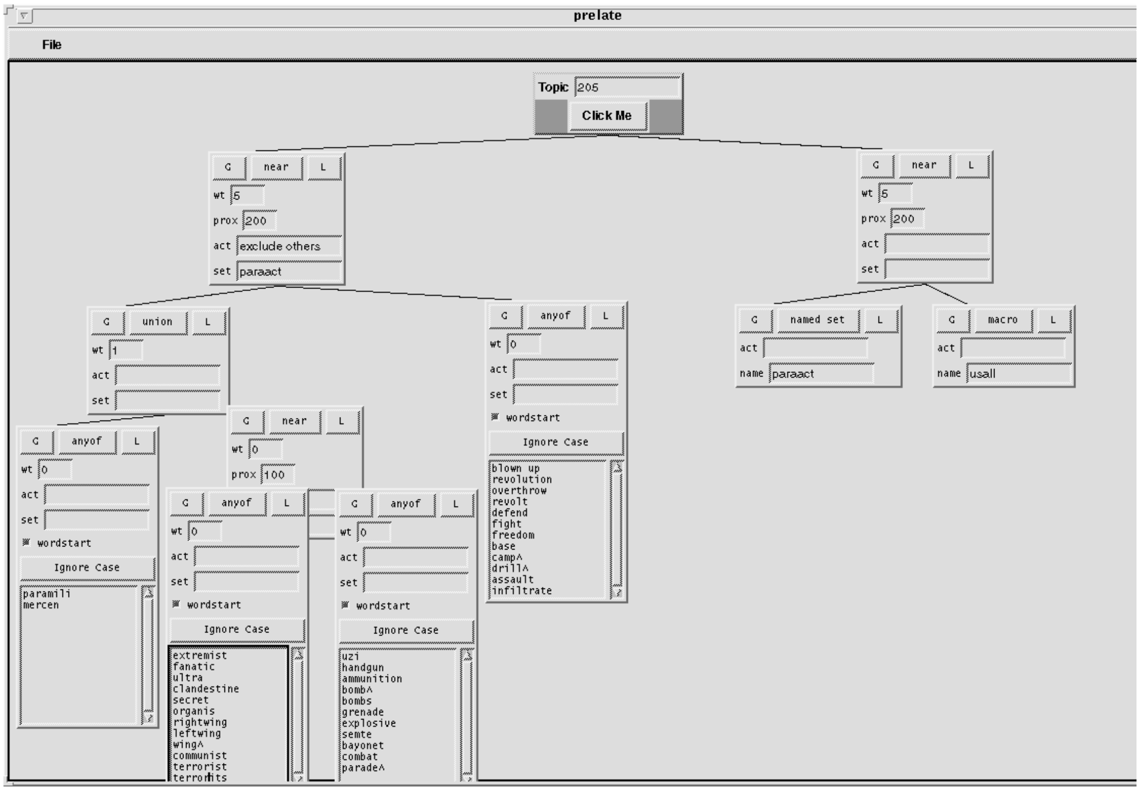

Figure 2: Screen dump of PRELATE representation of PADRE query for topic 205 relating to paramilitary activities in the U.S.A. The `usall` macro loads a pre-existing file of PADRE commands which define the U.S.A. concept in terms of alternative forms of the country name, state names, state capital names, names of geographical features, postal abbreviations, official state abbreviations and names of other prominent cities.

3. As for case 2, but where the same concept (tagged as a PADRE named set) may appear in more than one proximity relationship.

4. Cases in which an initial concept intersection, used to define a set of potentially relevant documents, itself becomes one component of a more restrictive concept intersection.

Figure 2 shows an example of multiple proximity relationships and the use of named sets.

3.2 Term Expansion Techniques

The expansion of terms to define a concept was done essentially by human free association supplemented by selective reference to the WordNet thesaurus, discussion with subject experts on some topics and by occasional references to a full lexicon of CD2/3. The latter was used to check whether a term occurred at all in the database, how it was spelled (eg. in Australian/British fashion as well as U.S.), and whether it was frequently mis-spelld in wayz which could be profitably included as search terms.

PADRE does not perform stemming. However, painful enumeration of all the different forms of a word was avoided through use of PADRE's ability to match arbitrary strings (optionally anchored at word starts). Regular expressions were used to locate dollar amounts in several queries.

3.3 Z-mode - Relevance Calculation Based On Proximity Spans

A relevance calculation method called, for want of better inspiration at 3am one morning just prior to the submission deadline, *Z-mode* was developed to more accurately reflect our working hypothesis (above). For each proximity relation instance discovered, the span of the relationship was calculated and the reciprocal of the span was added to the accumulated relevance score. The span is the length in words of the minimum substring of the text containing the instance of the proximity relationship.

Formula 1 above must be modified slightly to accommodate Z-mode. A document's estimated relevance R_d is now computed according to the following formula:

$$R_d = \sum_{t=1}^{k} (I_p \times r_{(p,d)}) \tag{7}$$

where:

k is the number of different proximity relationships in the query,

I_p is the manually assigned importance of proximity relationship p, and

$r_{(p,d)}$ is the relevance of a document d due to the specific proximity relationship p.

and:

$$r_{(p,d)} = \sum_{i=1}^{j} \frac{1}{S_i - 1} \tag{8}$$

where:

j is the number of instances of the proximity relationship found in document d, with unique starting point, and

S_i is the span of the ith instance.

For example, when searching the text below for a proximity relationship between *time, party,* and *people*, the italicised words constitute the first instance of such a relationship. The span of the relationship in this case is 15 and the contribution to the score of the document would be 1/14.

> The *time* has come for all good *people* to come to the aid of the *party.* We look forward to a time in which the people may party ...

The alert reader will notice that, assuming a typical proximity limit of 200 characters or so, the above example contains more than one instance of the sought-after relationship, including several with the same starting point. Overlapping instances are considered distinct provided they have unique starting points. Usually only the shortest of several spans sharing a common starting point is counted as an instance but, in some experiments, the longer ones were allowed to influence the score.

Initial trials with Z-mode on the training topics showed encouraging results. Subsequently, a number of variations on formula 8 were tried. Alternative numerators such as the total number of distinct proximity instances sharing the same start point or, alternatively, the same thing divided by the number of terms in the relationship were tried. Alternative denominators which reduced the rapid cutoff of the reciprocal were also investigated.

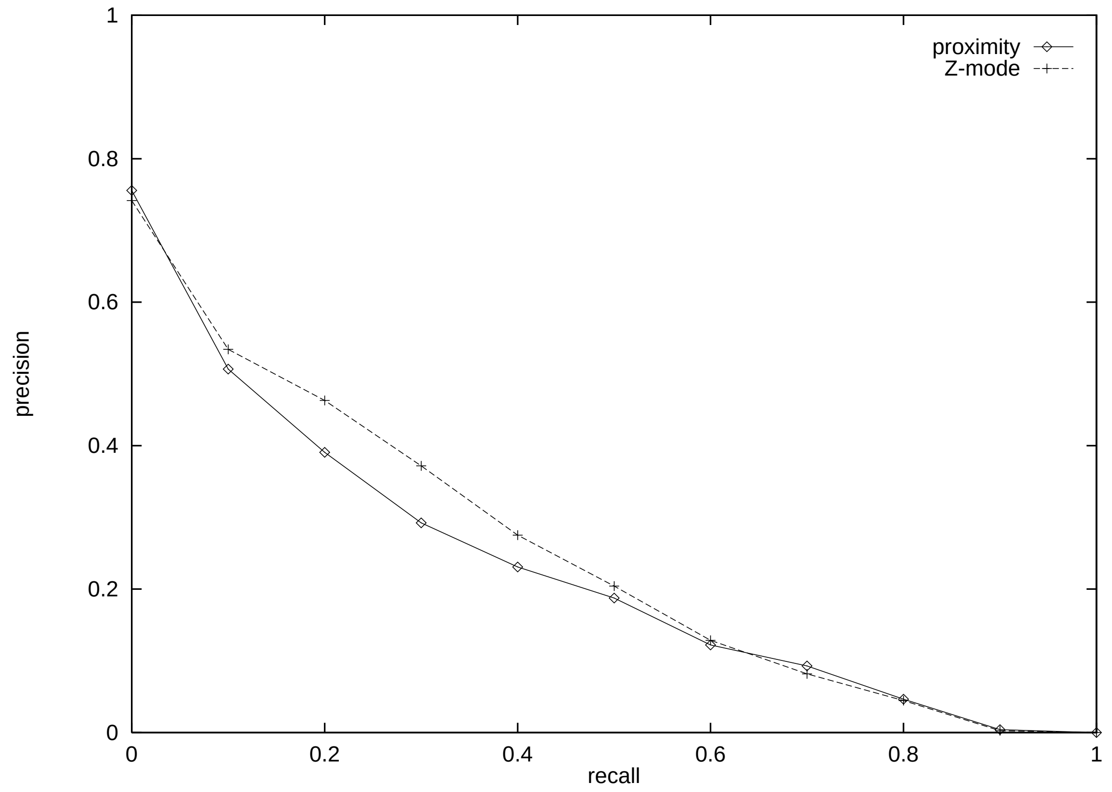

Figure 3: Precision-recall curves for PadreM1 and PadreZ.

On a dozen training topics, all worked better than the old non-Z-mode method but best results were achieved using the following formula:

$$r_{(p,d)} = \sum_{i=1}^{j} \frac{1}{\sqrt{S_i - 1}} \tag{9}$$

Note that Z-mode calculations of relevance do not involve collection frequency.

3.4 Experiment 1: PadreM1 (Initial Proximity-Only Run - Not Submitted)

A set of queries was written using the techniques described above and run using the weighting formula 3. Average precision over topics 202 - 250 was **0.2158**. The precision recall-curve is shown in figure 3.

3.5 Experiment 2: PadreZ (Official Manual Adhoc Run)

The padreM1 queries were manually modified in a systematic way. The Z-mode relevance function (formula 9) was selected. *Near* relationships searching for weightless loose phrases were left untouched but those which potentially contributed to relevance scores were changed to *znear* and the proximity limit was generally increased from 200 to 1,000 characters. This new set of queries, very strongly related to padreM1, was called padreZ.

Note that the padreZ results are significantly better than those for padreM1. Average precision has increased by 10% to **0.2383** and the total number of relevant documents retrieved has risen by 8%, from 3,326 to 3,602.

Compared with results obtained by other adhoc participants, padreZ:

- performed better than the worst for all measures on all topics.
- performed better than or equal to the median for:
  1. Rel. Retr. @ 100 on 32/49 topics
  2. Rel. Retr. @ 1000 on 23/49 topics
  3. Average precision on 26/49 topics
- achieved best performance for:
  1. Rel. Retr. @ 100 on 4/49 topics
  2. Rel. Retr. @ 1000 on 1/49 topics
  3. Average precision on 3/49 topics

3.6 Experiment 3: PadreW (Official Database Merging Run)

The padreZ queries were used unchanged in the database merging task. The ten subcollections were processed separately and the results combined by merging, sorting on relevance score, reranking and applying the 1,000 document cutoff.

As expected, the precision-recall results obtained are identical to those for padreZ. The only differences in the lists of documents retrieved were due to different orderings of equally ranked documents. Between the two lists of documents retrieved for all topics (each totalling 31,369 documents) only 3 differences were found. These all corresponded to a single query which produced a large group of equal scores around the 1,000 document mark.

3.7 Experiment 4: Enforcing a U.S.A. Context

At least five topics required rejection of otherwise relevant documents which did not relate to the U.S.A. In padreM1 and padreZ runs, this context was enforced by requiring the satisfying of (or increasing scores of documents which satisfied) an additional proximity test involving a very elaborate *usall* concept as described in figure 2.

In order to ascertain whether the considerable extra computational effort was justified by improved precision, the padreZ queries for the five above-mentioned topics were modified to remove the U.S.A. restriction and the results for the five new queries were compared with those for the originals.

When U.S.A. context was not enforced, average precision for the five queries fell from **0.3117** to **0.2840** and a significant difference in the precision-recall graphs may be seen in figure 4.

4 Automatic Query Generation

4.1 Experiment 5: PadreA (Official Automatic Adhoc Run)

The results for the automatic query generation run (padreA) were significantly inferior to our expectations, given the performance achieved during development against the TREC-3 topics and assessments.[2] Initial indications are that this poor performance was due, at least in part,

[2]Note that the TREC-3 topics were first translated to the terse TREC-4 style.

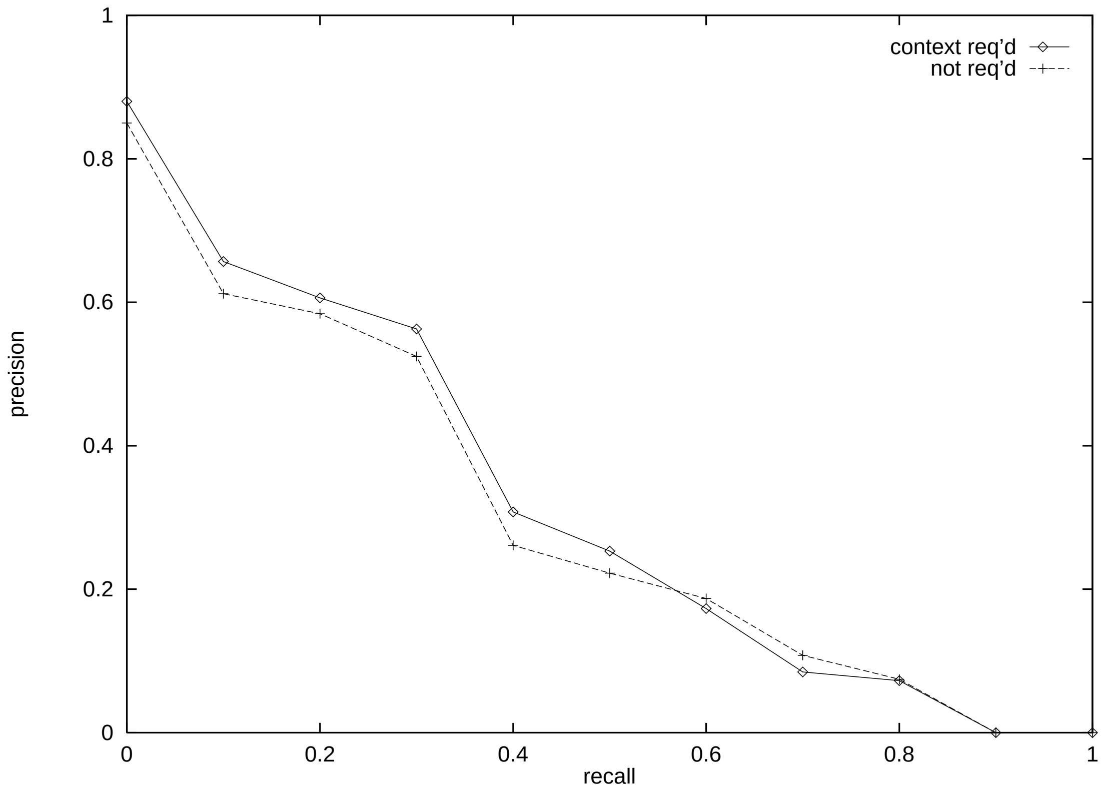

Figure 4: Precision-recall curves for queries 204, 205, 210, 227 and 235, showing the effectiveness of the method used to require a U.S.A. context.

to bugs in the mapping from the internal representation of the topic-query structure to the final PADRE query. The discussion of our automatic query generation approach will perforce focus on what was attempted rather than on what was achieved.

As with the padreZ run, our focus was on seeing to what extent proximity could be used as the primary ranking feature, rather than frequency.

The structure of automatically generated queries was two-fold: firstly, a set of anchoring terms was identified which was intended to define a subdomain on the document base with sufficiently high recall; then, subqueries derived from other parts of the topic were addressed to this subdomain of the document base, the purpose of which was to achieve better precision. Note however that the padreA queries used PADRE's standard `near` operator, rather than the `znear` used by padreZ, in order to bind these subqueries to the initial anchoring terms.

The terms in the padreA queries were restricted to being words or appropriate stems of words occurring in the topic. Additional capabilities permitting query expansion, using terms derived from relevance feedback and Wordnet thesaurus lookup, were not completed in time for inclusion. This was not considered to be a serious limitation to the experiment, as the primary motivation for query expansion would (we now believe incorrectly) be assumed to be to improve recall, and analysis of the TREC-3 topics and relevance judgements had shown that 99.4% of relevant documents contained at least one of the significant terms from the original topic. (A similar analysis of the TREC-4 data and gives a figure of 97.5%. A full discussion of these results will be the subject of another paper.) In retrospect, however, this restriction appears to be a significant limitation.

The poor performance of padreA manifested itself in two ways. In some cases, the software selected a set of anchoring terms that was too strict, and recall was low. In others, while initial recall was good, the subsequent subqueries did not make a sufficient improvement in precision.

Low initial recall was due to the software selecting the wrong terms from the topic to use as the anchor. This was in part due to the unwarranted assumption that the anchoring expression should be formed from a proper subset of the significant terms in the original topic. (The assumption is unwarranted because although a relevant document nearly invariably contains at least one word from the original topic, it need not of course be the same word for all documents relevant to the topic, and indeed in some cases best recall can not be achieved unless all significant words from the topic are incorporated into the anchoring expression.)

The second problem - poor precision - was probably due mostly to the restriction to using only terms contained in the original topic. Once our query expansion modules are complete, we should be in a better position to confirm this.

5 Discussion And Conclusions

Z-mode's total independence of collection statistics make it a very useful technique for database merging and routing, *provided* that:

- Z-mode queries are capable of achieving good precision-recall results, and
- Practical means for generating good Z-mode queries can be found.

On the first point, this year's Manual Adhoc results suggest that Z-mode queries are at least capable of achieving results which are better than average. Whether the reason they fall short of the very best relates to a fundamental limitation of Z-mode or whether it merely shows a lack of skill in query generation cannot be answered with present data. With the co-operation of the University of Waterloo we hope to investigate this question by converting their queries appropriately and processing with PADRE.

On the second point, manual generation of the queries was a time-consuming task, even with the help of PRELATE. We are optimistic that future ANU manual runs will involve only very small-scale manual intervention in an otherwise automatic process.

Acknowledgements

Fujitsu Laboratories provided access to AP1000 machines in the Fujitsu Parallel Computing Research Facility and assisted in various ways. Robin Stanton gave support and advice. Peter Bailey, Andrew Tridgell and David Campbell have worked on elements of the PADRE system. GNU regular expression code from the Free Software Foundation is incorporated in PADRE. We extend our sincere thanks to all these people and organisations.

Special thanks are also due to our subject experts and query advisers: Peter Bailey, Rosalie Balkin, Alan Connolly, Jo Evans, Michael Green, Sandy Gordon, Harold Griffiths, Kathy Griffiths, Ken Pogson, Maria Poulis, Tom Sutton, Andrew Tridgell, and David Westcombe.

The work reported has been supported by the Co-operative Research Centre for Advanced Computational Systems (ACSys).

References

[1] Fawcett, H. *PAT 3.3 User's Guide.* University of Waterloo Centre for the New Oxford Dictionary, Waterloo, Ontario (Feb 1991).

[2] Gonnet, G.H., Baeza-Yates, R.A. and Snider, T. *Lexicographic indices for text: Inverted files vs. PAT trees.* Report OED-91–01, University of Waterloo Centre for the New OED and Text Research, Waterloo, Ontario (Feb 1991).

[3] D.A. Hawking 'High Speed Search of Large Text Bases On the Fujitsu Cellular Array Processor,' *Proceedings of the Fourth Australian Supercomputing Conference*, pp. 83-90. Gold Coast, Australia (Dec 1991).

[4] D.A. Hawking and P.R. Bailey 'Towards a Practical Information Retrieval System For The Fujitsu AP1000,' in *Proceedings of the Second Fujitsu Parallel Computing Workshop*, paper P1-S. Kawasaki, Japan (Nov 1993).

[5] D.A. Hawking and P.B. Thistlewaite 'Searching For Meaning With The Help Of A PADRE', *Proceedings Of The Third Text REtrieval Conference (TREC-3)*, pp. 257-268. US Department of Commerce, NIST Special Publication 500-225 (Apr 1995).

[6] D.A. Hawking 'PADRE — A Parallel Document Retrieval Engine', in *Proceedings of the Third Fujitsu Parallel Computing Workshop*, paper P2-C. Kawasaki, Japan (Nov 1994).

[7] D.A. Hawking and P.R. Bailey 'A Document Retrieval Architecture Supporting Terabyte Collections', in preparation.

[8] D. Hawking, P. Bailey, D. Campbell, P. Thistlewaite and A. Tridgell 'A PADRE in MUFTI (A Multi User Free Text retrieval Intermediary)', in *Proceedings of the Fourth Parallel Computing Workshop* paper 26, Imperial College, London, (Sep 1995).

[9] Horie, T., Ishihata, H., Shimizu, T. and Ikesaka, M. 'AP1000 Architecture And Performance Of LU Decomposition,' in *Proceedings of the 1991 International Conference On Parallel Processing*, pp. 634-635. August 1991.

[10] Ishihata, H., Horie, T., Inano, S., Shimizu, T. and Kato, S. 'CAP-II Architecture,' in *Proceedings of the First Fujitsu-ANU CAP Workshop*, paper 1. Kawasaki, Japan (Nov 1990).

[11] A. Tridgell and D. Walsh 'The HiDIOS Filesystem' in *Proceedings of the Fourth Parallel Computing Workshop*, pp. 53-63. Imperial College, London (Sep 1995).

[12] *PADRE home page.* `http://cap.anu.edu.au/cap/projects/text_retrieval/`