TREC 14 Enterprise Track at CSIRO and ANU

Mingfang Wu[1] Paul Thomas[2] David Hawking[1]
mingfang.wu@csiro.au, paul.thomas@anu.edu.au, david.hawking@csiro.au

[1] CSIRO ICT Centre, Australia
[2] Department of Computer Science, Australian National University

1. Introduction

The primary goals of the CSIRO and ANU team's participation in the enterprise track were two-fold: 1) to investigate how well our search engine PADRE responds to the new collection and the new tasks, and 2) to explore if document structure specific to an email collection can be used to improve system performance.

By the time of submission deadline, we completed two tasks: known-item search and discussion search. For both tasks, we used the PADRE retrieval system [1], in which the Okapi BM25 relevance function was implemented. Each message in the collection was treated as an independent document, so both topic distillation scoring and same site suppression mechanism were turned off (i.e. -nocool and –SSS0 respectively). During the indexing, stemming and stopword elimination were not applied and sequences of letters and/or digits were considered as indexable words.

We parsed the HTML pages in the original collection into an XML format (the DTD is shown in the appendix), and removed non-email pages. Our parsed collection includes 174,311 email messages, and we used this collection for our experiments.

2. Known-item search task

The known item search task is aimed at finding an important email that is known to exist. We tried seven runs and submitted five runs for this task. Table 1 summarises the indexing and the retrieval environment for the five submitted runs.

Table 1: Indexing and retrieval settings for the known-item task

Run ID	Index	Query	Weighting
csiroanuki1	The email structure was ignored and all elements were treated as content.	Title	Okapi BM25
csiroanuki2	The email structure was used: from, subject, to, cc and date were indexed as metadata; the texts in all other elements were treated as content.	Title	Okapi BM25 + subject text is up-weighted

csiroanuki3	Similar to csiroanuki2, but quoted and forwarded message fragments and signature were ignored and excluded from the content.	Title, with times and names modified (see below)	As in csiroanuki2
csiroanuki5	As in csiroanuki3	Title	As in csiroanuki2
csiroanuki6	Similar to csiroanuki3, but the subject text was repeated to promote the importance of subject.	Title	As in csiroanuki2

To make use of metadata in the run csiroanuki3, we transformed two types of queries. 1) Queries containing a name: whenever a person's name was detected in the topic title, the person's name was quoted and his/her email address(es) was added to the query. For example, the title of the topic 69 *official introduction to Dan Connolly*, was transformed to *official introduction to "Dan Connolly" [a:connolly@hal.com a:connolly@www10.w3.org a:connolly@w3.org]*. According to PADRE's query language, this query would be transformed internally to: retrieve the email that contains one or more terms from "official introduction to 'Dan Connolly'" with connolly@hal.com, connolly@www10.w3.org or connolly@w3.org in the "author" metadata class (i.e. the "from" element). 2) Queries containing dates: the abbreviation of the month was added. E.g. the title of topic 139, *W3C talks in April 2004*, was transformed to: *w3c talks in [apr april] 2004*.

Table 2 System performance for the known-item task

Run ID	MRR (gain)	S@10	Fail@100
Csiroanuki1	0.468 (0%)	94 (75.2%)	11 (8.8%)
Csiroanuki2	0.502 (7%)	96 (76.8%)	13 (10.4%)
Csiroanuki3	0.515 (10%)	96 (76.8%)	13 (10.4%)
Csiroanuki5	0.522 (12%)	97 (77.6%)	14 (11.2%)
Csiroanuki6	0.504 (8%)	96 (76.8%)	14 (11.2%)

Table 2 shows the performance of each run. In terms of the MRR measure, all the test runs csiroanuki2, 3, 5 and 6 are better than the base run csiroanuki1 (among them, csiroanuki3 and csiroanuki5 are significantly better than the run csiroanuki1 ($p < 0.04$ for the paired, two tailed t-test)). The run csiroanuki5 achieved highest MRR while a simple strategy, ignoring the quoted and forwarded text and up-weighting the subject text, was adopted here. Nevertheless, the MRR of csiroanuki5 is lower than the median of all participants' runs by 8%.

The known-item message appears in the top ten for more than three quarters of the topics. All runs tend to succeed and fail at the same topics. For example, all runs failed to get the known-item for the same set of ten topics.

3. Discussionsearchtask

The discussion search task was to search for messages pro and con in an argument/discussion regarding to a topic. Again we submitted five runs (from nine). The indexing and the retrieval environment for each run are shown in Table 3.

Table 3: Indexing and retrieval settings for the discussion search task

Run ID	Index	Query	Weighting

csiroanuds1	The email structure was ignored and all elements were treated as content	Query	BM25
csiroanuds3	The email structure was used: from, subject, to, cc and date were indexed as metadata; the quoted text, forwarded text and signature are all ignored	Query	BM25 + subject is up-weighted
csiroanuds5	As in csiroanuds3	Disjunctive query	As in csiroanuds3
csiroanuds7	As in csiroanuds3	Query	As in csiroanuds3
csiroanuds8	As in csiroanuds3	Query + expanded query	As in csiroanuds3

Csiroanuds1 is our base run; its setting is very much like the csiroanuki1 in the above known-item task. As in the discussion search task "a correct answer is an email which contributes a pro or con relating to the topic, in new (not quoted) text", we ignored the quoted text, forwarded text and the signature for the run 3, 5, 7 and 8.

In runs csiroanuds1, 3, and 7, we took the text from the query field of a topic as a query, and made minimal modifications to use PADRE's syntax. (For example, for the topic DS33, *plug-in patent*, our fed query is *"plug-in" patent* as our search engine tends to separate the two words *plug* and *in* if they are not quoted.) Csiroanuds5 used a disjunctive form: the query was constructed as a disjunction of terms to attempt to increase recall at the expense of precision.

As the discussion task encourages high recall, in the run csiroanuds8 we adopted the traditional pseudo-relevance feedback algorithm for query expansion according to the following steps:

1) An initial list of ranked message was obtained by using the original query field of a topic;
2) All terms in the first ten documents were ranked according to the following term selection value:

$$TSV = w^{(1)} * r/R$$

The weight $w^{(1)}$ is the Robertson/Sparck Jones weight:

$$\log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5) / (N-n-R+r+0.5)}$$

where r is the number of messages that contain the term, R is the number of selected messages, n is the number of documents containing the term, and N is the number of messages in the collection.

3) The top 20 terms that were not in the original query were added to the original query, and the new added terms were down weighted by a factor of 3.

Table 4, Figure 1 and 2 show the performance of all runs. Overall, all the test runs (csiroanuds3, csiroanuds5, csiroanuds7 and csiroanuds8) are worse than the base run csiroanuds1

in terms of any measure. The common difference between the four testing runs and the base run is that the quoted and forwarded text are ignored in the four testing runs as we thought that might help to retrieve new text, obviously this strategy does not help. Probably we should have kept quoted and forwarded text in those messages that have new text.

Considering the performance next best to the base run is run csiroanuds7, and the other three test runs have worse performance than run csiroanuds7, the query and weighting variations in these three runs (e.g. expanded query or disjunction query, up-weighting the subject text) do not appear to help here either. After the evaluation judgments are released from NIST, we could test if these query and weighting strategies work for the index method as in run csiroanuds1.

Table 4. Average precision and BPref for the discussion task

Run ID	AveP (gain)	BPref (gain)
csiroanuds1	0.319 (0%)	0.331 (0%)
csiroanuds3	0.286 (-10.3%)	0.308 (-6.8%)
csiroanuds5	0.253 (-20.7%)	0.269 (-19.5%)
csiroanuds7	0.297 (-6.9%)	0.321 (-2.8%)
csiroanuds8	0.259 (-18.8%)	0.283 (-15.1%)

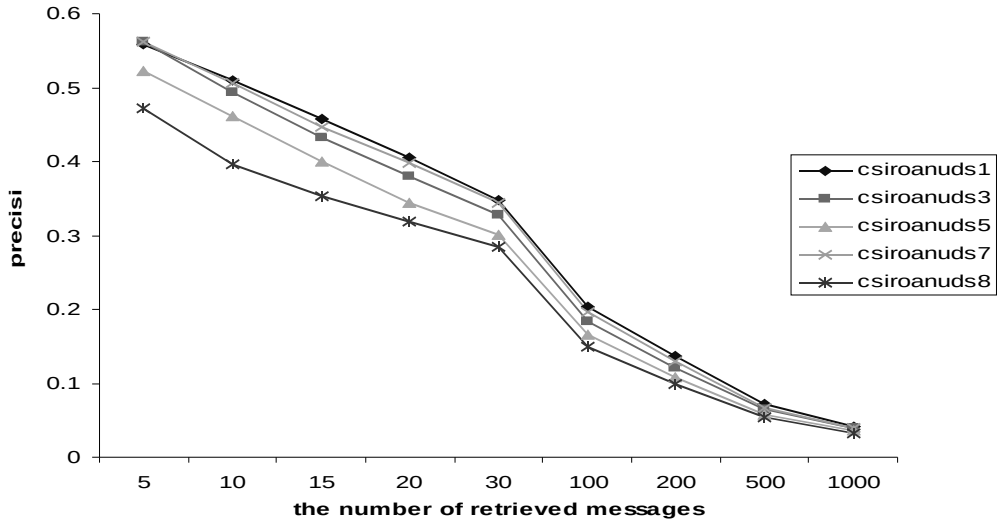

Figure 1. The precision at cut-offs for discussion search task

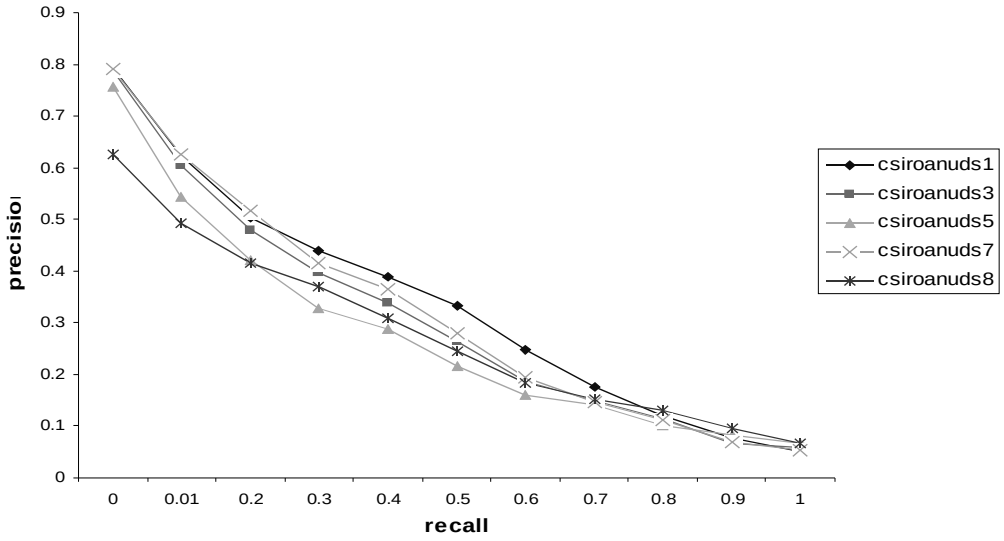

Figure 2. Interpolated precision for the discussion task

4. Discussion

For known-item search, a simple strategy, ignoring the quoted and forwarded text and up-weighting the subject text, achieved the highest MRR. For the discussion search task, all our test runs performed poorer than the base run that simply ignored email structure and treated all elements equally. Maybe there are bigger gains to be made from considering email-specific features like thread structure.

We observed that the format of the email archives (in HTML) has caused some difficulties to the task of email retrieval. This has led to some groups returning non-email pages for judging. We expect that should be easier next year though now that several teams have parsed the HTML and extracted email from it.

References

[1] Hawking, D., Craswell, N., and Thistlewaite, P. ACSys TREC-7 Experiments. In Proceedings of Seventh Text Retrieval Conference. November 1998. Gaithersburg, USA.

[2] Robertson, S. E., Walker, S. and Beaulieu M. Experimentation as a way of life: Okapi at TREC. In *Information Processing and Management* 36(2000) pp.95-108

Appendix: DTD for email representation

```
<!-- a thread is a list of messages, ordered from oldest to newest -->
<!ELEMENT thread (message|placeholder)+>

<!-- a placeholder is just a token saying we know there's a message here, -->
<!-- but not what it is -->
<!ELEMENT placeholder EMPTY>

<!-- header fields in parsed message -->
<!ENTITY % headers "document-id?, to?, from?, cc?, subject?, date?,
                    message-id?, in-reply-to?, references?, reply-to?,
```

```
                timestamp?, name?, parse-info?">
<!ENTITY % body "original?, quoted?, forwarded?, signature?, url*,
            attachment*">

<!-- a message is the real deal -->
<!ELEMENT message (%headers;, %body;)>
<!ATTLIST message id CDATA #REQUIRED>

<!-- various headers -->
<!ELEMENT document-id (#PCDATA)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT subject (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT message-id (#PCDATA)>
<!ELEMENT in-reply-to (#PCDATA)>
<!ELEMENT references (#PCDATA)>
<!ELEMENT reply-to (#PCDATA)>
<!ELEMENT timestamp (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT parse-info (#PCDATA)>

<!-- original parts only -->
<!ELEMENT original (#PCDATA)>

<!-- quoted parts only -->
<!ELEMENT quoted (#PCDATA)>

<!-- forwarded parts only -->
<!ELEMENT forwarded (#PCDATA)>

<!-- signature block only -->
<!ELEMENT signature (#PCDATA)>

<!-- extracted URLs -->
<!ELEMENT url (#PCDATA)>

<!-- attachments; they can be inline with a given file name, or -->
<!-- stored in a named external file                          -->
<!ELEMENT attachment (#PCDATA)>
<!ATTLIST attachment filename CDATA #IMPLIED

<!-- this is an attachment, stored externally -->
<!ELEMENT ext-attachment (#PCDATA | to | from | cc| subject | date |
                  timestamp | name)*>
<!ATTLIST ext-attachment id CDATA #REQUIRED
                    filename CDATA #REQUIRED>
                  external CDATA #IMPLIED>
```