Efficient and Flexible Search Using Text and Metadata*
CSIRO Mathematical and Information Sciences
Technical Report 2000/83

David Hawking[a], Peter Bailey[b] and Nick Craswell[a]
[a]CSIRO Mathematical and Information Sciences
GPO Box 664,
Canberra, Australia
{David.Hawking, Nick.Craswell}@cmis.csiro.au
[b]Department Of Computer Science
Australian National University, Canberra, Australia
Peter.Bailey@cs.anu.edu.au

May 2000

Abstract

Digital libraries and intranets often include a wide mix of document types, ranging from catalog entries consisting entirely of metadata, to documents with both substantial text content and thorough metadata tagging, to documents with hardly any metadata at all. This paper describes a retrieval model for use in such environments which can accommodate searches in a variety of metadata schemata. It uses both content search and three-valued satisfaction of metadata constraints (yes, no and maybe). In this model queries may include explicit metadata constraints as well as free text terms. The latter may impose additional implicit constraints as well as contributing to content relevance scores. Result lists are presented in tiers, in which the top tier corresponds to full satisfaction of all constraints and subsequent tiers are derived by progressive relaxation of constraints. An efficient and scalable implementation of the model has been developed and has been in production use on the Australian National University intranet since July 1999. Details of the implementation are given and performance results are presented.

KEYWORDS: Information retrieval, search, metadata

1 Introduction

Document retrieval in digital libraries and, increasingly, in government and corporate intranets is best served by a combination of metadata processing and content searching.

It is easy to think of information needs which cannot be satisfied with either alone. Searchers must rely on content if metadata is absent, erroneous or incomplete. Unfortunately, large legacy collections combined with budgets insufficient to permit complete and consistent tagging mean that this is often the

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

case. Content search is also required when extensive metadata is present, since it is extremely unlikely that metadata subject tags can ever completely capture every nuance of the content of a document.

On the other hand, in museums, libraries and repositories, some or all of the documents may be catalog records consisting entirely of metadata. Free-text searching is not suited to this environment at all. In any case, without recourse to metadata, searching will do a very poor job of identifying documents published by Penguin in 1998, or of distinguishing documents written by Bill Clinton from those written about him.

This paper describes a new model for combining constraint satisfaction with content searching in a way which:

1. Takes into account missing, uncertain or inconsistent metadata,
2. Allows graceful degradation if not all constraints can be satisfied,
3. Presents results in an order which has validity with users,
4. Allows users to specify the relative importance of constraints and terms.
5. Reduces loss of potentially useful search information during indexing.
6. Can be implemented efficiently and scalably on low-cost hardware.

The design has been substantially implemented and the result is in production service on the Australian National University intranet [2]. In the present implementation, metadata is assumed to be contained within the document (*internal metadata*). However, the extension to handle external metadata is straightforward, at least logically.

The present paper is concerned only with the problem of retrieving documents in a digital library, using content, tags, captions or transcripts which are in the text domain.

In general, digital libraries will be both distributed across multiple servers within the hosting organisation and will rely heavily on links to external information services. However, issues relating to resource discovery within the organisation (eg. spidering and other methods) and to distributed information retrieval are beyond the scope of the present paper, as are techniques to cope with errors in text arising from OCR scanning or speech recognition.

2 Related work

2.1 Content ranking of a complete-match answer set

A number of contemporary commercial database management systems (such as Oracle and Informix) support relevance ranking of textual fields in a database. However, in these systems, as far as the present authors are aware, queries which combine a relational search with content ranking give absolute priority to the relational attributes. That is, the only records which are ranked are those which completely satisfy the relational constraints.

De Fazio et al [4] and Fontaine [5] describe the addition of full-text ranking to, respectively, the Oracle and POSTGRES database systems. De Fazio et al [4] note the need for database query languages to include the element of uncertainty.

Currently, certain Web search engines (such as Alta Vista) allow query results to be restricted in a similar way. For example, a specification of `host=www.netscape.com` will return only the results for which the condition applies.

In the model proposed here, the results produced in the above type of system are extended by a series of further result tiers corresponding to progressively weaker degrees of partial match.

2.2 Incorporation of uncertain facts in a probabilistic retrieval model

The model proposed by Fuhr [7] treats the relational constraints (fact conditions) and the text query terms (text conditions) in a uniform way. Fact conditions are included in the probabilistic structure of the text retrieval system. For a particular document, real-number weights are assigned to each of the conditions, both factual and textual. Text condition weights are assigned in the normal IR way and fact condition weights are estimates of the degree of match. In an example given by Fuhr, the degree of match for an `author=` condition is determined by the proportion of common trigrams between the query specification and the author attribute of the document. Similarly, the degree of match for date conditions depends upon the numerical difference between the specified and actual dates.

Later work by Fuhr and Rölleke [8, 15, 17, 9, 16] has extended this research to take account of document structure, hyperlinks and multi-media on one hand and four-valued logic (to account for missing and inconsistent fact information) and user-weighted query elements on the other.

The model proposed here is much less general than those of Fuhr and Rölleke but is oriented toward efficient implementation. In our envisaged applications, we do not see a need to distinguish between “unknown” and “inconsistent” truth values.

Barja et al [3] describe Informia, a system for integrating large numbers of heterogeneous search resources. Their system appears to accommodate combinations of metadata constraints and content terms but the focus of the paper is on mediating searches across heterogeneous search services and on correctly merging the results obtained.

2.3 Relation to the TREC ad hoc retrieval task

Experience in TREC ad hoc [18] suggests that information needs such as those of a college student preparing a term paper, are well handled by systems which use word-stemming, case-folding, stopword elimination, query expansion and careful weighting of query and document terms to process lengthy natural language queries. On average, documents are found to be well ranked using a weighting formula which sums the contributions due to each query term, taking into account frequency of the term in the query, frequency of the term in the document, proportion of other documents also containing the term, and the length of the document.

We maintain that techniques such as word-stemming, case-folding and stopword elimination can be very useful during query processing but, if applied during indexing, may discard information which could enable more precise querying. Examples include distinguishing Ford cars from river crossings, distinguishing “the Pope” from all the people called Pope and maintaining the important distinctions between the proper nouns “Woods” and “Wood” and the common words “wood”, “wooden” and “wooded”. Such discriminations are believed to be much more important in Web or Intranet search, where information needs are mostly different from those modelled in TREC ad hoc, and where queries are much shorter.

Most of the best performing retrieval systems on the TREC ad hoc task, compute relevance scores for documents by summing the weights associated with each query feature present in the document. A high relevance score may be computed for a document containing only a subset of the query features. For example, the query “vitamins and health” may rank documents containing many vitamin references above some documents satisfying the whole query. Looking at a results list, the user may be quite baffled as to why certain high-ranking documents have nothing to do with health.

By contrast, a purely Boolean retrieval system is easily auditable. Every document retrieved can (if the system is working correctly) be seen to satisfy the stated conditions.

The present proposal attempts to retain the auditability of the Boolean system without completely losing the advantages of the term-weighting schemes. For short queries, all the documents which fully satisfy the conditions will be ranked above others which do not. However, because Boolean queries are sometimes too strict (or do not perfectly represent the user’s intention) and thus produce incomplete result sets, further groups of results may be included in the ranking in the order determined by a progressive relaxing of constraints. In other cases, the number of full Boolean matches is large and the content ranking of the top tier of results is of benefit.

3 Proposed retrieval model

A query $\mathcal{Q}$, consists of an ordered set $\mathcal{A}$ of *attribute constraints* $\{A_i : 0 \leq i < m\}$ merged with a set $\mathcal{C}$ of *content constraints* $\{C_j : 0 \leq j < n\}$.

The order of the attribute constraints is taken to signal the relative priority given to them by the user. By contrast, the relative importance of the content constraints is understood to be taken care of by the relevance scoring formula.

Some queries consist of a large number of content terms. This may arise when a searcher pastes paragraphs from an interesting document into the query window of the search system. In such cases, it is counter-intuitive to treat every content term as a constraint. Instead, the terms are treated disjunctively and document ranking is on the basis of relevance score.

For the purposes of defining the model, what constitutes an attribute constraint and what constitutes a content constraint are unimportant. However, intuitively, constraints may require that the publication date may lie in a specified range or that the subject tag should include the string `Clint`. Similarly, content constraints may potentially include phrases, proximity relations, words, arbitrary substrings, or sets of these things.

$\mathcal{Q}$ is evaluated over a set of documents $\mathcal{D}$ in order to produce a list $\mathcal{R}$ of the best documents, ranked according to the method described below.

Note that either the constraint set or the term set may be empty, corresponding to pure metadata queries and pure content queries respectively. The case where both sets are empty is not very interesting!

3.1 Document accumulator structure

To support the method, each document is considered to be associated with the following three accumulators:

Y An integer in which the ith bit is set iff the ith query component (either constraint or term) has definitely been satisfied.

M An integer in which the jth bit, is set iff metadata fields corresponding to the jth pre-defined metadata class are missing, or contain inconsistent values. The setting of these bits is not query dependent. In the applications envisaged, there does not appear to be an advantage in distinguishing missing attributes from inconsistent ones. At query time a mapping must be made between query constraints and the corresponding bits in M.

S A number representing the document's content score as calculated using the Okapi BM25 formula, [12, 14] modified to eliminate the possibility of negative score contributions.

$$lf = \begin{cases} \frac{N-n+0.5}{n+0.5} & \text{if } N > 2n \\ 1.0001 & \text{otherwise} \end{cases}$$

$$w_t = q_t \times tf_d \times \frac{\log(lf)}{2 \times (0.25 + 0.75 \times \frac{dl}{avdl}) + tf_d}$$

where w_t is the relevance weight assigned to a document due to 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).

3.2 Query processing

Each query component is processed in order.

1. The appropriate Y or M bits are set for each document which, respectively, matches *or* neither matches nor violates the constraint, and

2. If it is a content constraint, the value of S for each such document is increased by the appropriate amount.

3. It is not yet clear whether metadata constraints should update S or merely Y and M. In the initial implementation, non-numeric (ie. non-date) metadata do update the value of S, using the Okapi formula. For example, the metadata constraint `m:Simon` updates the S value for each document which contains an occurrence of Simon in a metadata field mapped to the “m” metadata class. The value of n used in the Okapi formula is the number of documents which contain an occurrence of Simon in a metadata field mapped to the “m” metadata class.

   If experience confirms that this approach is justified, consideration will be given to extending the implementation to treat dates in the same way.

3.3 Result ranking

The candidate result set comprises all documents which have at least one Y, or M bit set or which have non-zero S. This set is sorted to produce $\mathcal{R}$ using the following sort keys, ranked in order of decreasing primacy.

k1 The number of bits set in Y,

k2 The number of bits set in M,

k3 For queries including metadata constraints, the pattern of Y bits,

k4 For queries including metadata constraints, the pattern of M bits,

k5 The value of S.

k6 Document order within the collection.

Keys k1 and k2 implement ranking by the extent to which documents satisfy the constraints, including the content constraints. Keys k3 and k4 are not used in content-only queries. When attribute constraints are used however, the order of the constraints in a query is interpreted as a priority order. Key k5 implements normal content ranking and k6 merely specifies what happens when there is no other way to distinguish two documents.

3.3.1 Result tiers

Rankings resulting from sorting on the basis of keys k1, and k2 form successive (possibly empty) tiers. The first tier comprises the documents which satisfy all $|\mathcal{Q}|$ constraints. The second comprises those which definitely satisfy $|\mathcal{Q}| - 1$ of them and may possibly satisfy the other one. The third tier comprises those documents which definitely do not satisfy one constraint but which satisfy all the others.

Within tiers defined in this way, documents are ranked according to their accumulated relevance score (k5).

Tiered ranking is illustrated in Figure 1.

```
QUERY: a:russell chemistry d=jun1999

TIER 1 RESULTS
1.  (3, 0, 85\%) http/chemserver.anu.edu.au/rab/research/index.html
     Research Interests - 2 jun 1999
2.  (3, 0, 83\%) http/rsc.anu.edu.au/\%7Earussell/research.html
     Research - 22 jun 1999

TIER 2 RESULTS
3.  (2, 1, 49\%) http/rsc.anu.edu.au/\%7Etorda/chemists.html
     Chemistry Odds and Ends - 17 jun 1999
4.  (2, 1, 47\%) http/rsc.anu.edu.au/\%7Earussell/favlinks.html
       - 22 jun 1999
...

TIER 3 RESULTS
26.  (2, 0, 100\%) http/rsc.anu.edu.au/\%7Earussell/index.html
      Anthony J. Russell homepage - 19 mar 1999

TIER 4 RESULTS
27.  (1, 2, 55\%) http/www.anu.edu.au/pad/pubs/handbooks/1998/ug/98chem.html
      ANU Undergraduate Handbook 1998 - Chemistry -
```

Figure 1: A sample web search query and its results. The numbers in parentheses indicate: the number of yes bits, the number of maybe bits and this documents content score as a percentage of the highest for any document. In the tier 1, all constraints are satisfied and results are ranked by content score. Tier 2 contains 23 documents which match two of the constraints and have a MAYBE value for the third. Sub-tiering is not used because it happens that all results satisfy the same constraints. Consequently this tier is also ranked by content. Tier 3 contains the document with the highest content score. It satisfies the author and content constraints but violates the date constraint. Tier 4 starts with documents whose content references chemistry but whose author and date information is missing or inconsistent.

3.3.2 Sub-tiers reflecting the relative importance of constraints

For queries including attribute constraints, the tiers described in the previous section may be too coarse grained. In this case, it is desirable to divide the tiers into sub-tiers reflecting the relative importance of the constraints to the user. Keys k3 and k4 are selectively applied to achieve this.

For example, if a query requested both that the author should be `Moffat` and the date `1990`, the ordering of documents which satisfied only one of these constraints would be determined by the order in which the user had listed them. If the author constraint appeared first, then documents by Moffat published before or after 1990 would be ranked ahead of documents published in 1990 by other authors.

Constraint order is not used to separate results differing only in which constraints achieved MAYBE values.

The use of sub-tiers does not make sense either for queries with only content terms, such as those used in TREC Ad Hoc tasks [13]. In such cases, the relevance estimation formula is likely to do a much better job than the user in ranking the importance of the terms and sorting keys k3 and k4 are ignored.

3.3.3 Disjunctive groups

As described so far, the retrieval model is essentially conjunctive although the union of all the tiers represents the disjunction of the constraints. Explicit disjunction is provided by left and right truncation operators, by a stemming operator and by the use of square brackets to indicate disjunctive groups of terms and/or constraints. For example, `Hawk*` (right truncation) represents the disjunction of all words starting with Hawk and `hawk eagle` represents a constraint satisfied by the presence of either hawk or eagle.

3.4 Treatment of negation

Both content and attribute constraints may be negated. A negated constraint is just another constraint. See Section 5.2.2 for examples. A negated content constraint, does not affect content scores.

3.5 Mandatory constraints

Sometimes searchers feel certain that no document which fails to match (or which matches) a particular constraint is worth looking at. For these circumstances, the `+` and `-` operators are provided. No document which fails to satisfy a constraint of this type will appear in any tier of the final ranking.

3.6 Which parts of the document comprise the content?

Users should be able to specify whether content searches should look at the entire document (including metadata) or be restricted to the non-metadata parts of the document. In the current design, the non-metadata parts of the document and each separate metadata class are indexed separately, but a universal metadata class is available which subsumes all the metadata fields of the document as well as the non metadata content.

Users expect that title, and possibly keyword, metadata should be considered both as part of the document content and also as part of the appropriate metadata class. This functionality is not yet implemented but the index design (Section 7.4) will support it efficiently. It is unclear whether the value of n in the Okapi formula should be appropriate to each type of occurrence or aggregated across content, title and keywords metadata classes. If the former were to apply, occurrences of a query word in a title would usually count much more than occurrences in the body of a document.

4 Treatment of multiple metadata schemata

The design of the retrieval system accomodates the use of multiple metadata schemata by providing a set of plug-in lexical scanners, one for each schema. At present, documents conforming to different schemata may not be included within the same index. However, the query processing design allows queries to be processed over multiple indexes, thereby potentially allowing searching over a collection with heterogeneous schemata.

Mapping between elements of different schemata is achieved through *metadata classes.* Value data from one or more different metadata fields is mapped to a single metadata class, represented by a single lowercase letter. In the email search example below, text on the To, Cc and Bcc lines is all considered part of the "t" metadata class. Across sets of documents representing different metadata schemata, metadata classes with similar meaning should be mapped to the same letter. If both the From lines in email messages and the author metadata values in web pages are mapped to metadata class "a" then searches across a merged collection of email and webpages can make use of basic metadata search.

Note that the treatment of metadata in the present system is very rudimentary and designed to support searching only. The goal is to extend basic content-only search to allow simple relational constraints such as author, publisher and date. No pretence is made that this approach is yet suitable for complex, structured or spatial metadata repositories.

5 Case study 1: Searching email archives

Simple email messages in the Unix environment consist of a series of lines of text. The header of a message is delimited from the body by a blank line. Some header lines start with reserved strings (such as `Subject:`, `Date:`, and `From:`), indicating the start of various types of metadata about the message. The body constitutes the message itself.

The **searchmail** application implements our retrieval model, allowing a person to search archives of email by posing queries which include both content and metadata constraints. For each query, searchmail collects result messages in a temporary mailbox for viewing by the person's email reader.

The philosophy of searchmail is that, when few or no messages match the full query, partial results obtained by successively weakening the constraints should be presented. Perhaps the wanted message was sent by someone else, or at a different time or perhaps it misspelled the desired keyword in the subject line.

5.1 Content-only queries

The following examples illustrate searchmail content-only queries. Note that in the third and fourth examples there is an implied AND between the terms but that partial results are also presented in a tiered way, reflecting progressive weakening of the AND condition.

1. `Clinton` Will find only messages containing the word Clinton (or clinton) and rank them by their Okapi score.

2. `Clint*` Will find messages containing words starting with Clint or clint.

3. `Clinton mail Lewinsky` Messages containing all three terms will be ranked ahead of all others. The word mail is likely to be less rare than either of the others, and thus to contribute less to the ranking of these messages.

4. `[UK ‘‘United Kingdom’’ Britain]` In this case, the three terms (the phrase “United Kingdom” counts as a single term) are grouped (implied Boolean OR) and documents containing all three terms are not in general ranked ahead of others.

5. `[UK ‘‘United Kingdom’’ Britain] industr*` Documents containing both a word starting with `industr` and at least one of the group of three are ranked ahead of others.

5.2 Metadata-only queries

The following examples illustrate searchmail queries using metadata only. A single letter followed by a colon and a search expression, indicates that the search is to be applied only to the metadata line indicated by the letter (f - from, t - to, s - subject, etc). The date field (D) is handled as a numeric quantity and is tested using arithmetic relational operators such as “less than”.

6. `f:"Ned Kelly"` Find messages which include the phrase “Ned Kelly” in the From line.

7. `[f:Ned f:Kelly]` Find messages which include the word Ned or the word Kelly in the From line.

8. `f:Beatty d>10Jan` Find messages which include the word Beatty in the From line and which are dated later than the 10th of January (of the current year). Messages which satisfy both constraints will be ranked in the top tier (and will be ranked in the order in which they appear within the mail archive since no content query was specified). Because of the query order, messages which satisfy the from condition only will be ranked ahead of messages which match the date condition only.

9. `f:Templar s:aard* d>10Jan97<10Jan98` Note the use of right-truncation in the subject line constraint and the use of the double relational operator in the date constraint (treated as a single constraint). Messages satisfying all three constraints will be ranked in the top tier. Messages satisfying only two constraints will be ranked ahead of those satisfying only one, and, because of term order in a metadata-only query, those satisfying f and s conditions will rank ahead of those satisfying f and d, and those satisfying s and d.

5.2.1 Queries combining content and metadata

10. `‘‘National Library of Australia’’ f:Dack` Messages which include the phrase “National Library of Australia” and include the word Dack in the from line will be ranked first. Because of the query order, messages which contain the phrase “National Library of Australia” but do not satisfy the from condition will rank ahead of messages from Dack which do not include the phrase. Within each sub-tier documents will be ranked by content score.

11. `Clinton [Lewinsky Jones Flowers ] f:Starr d>19Jan1998` Messages from Starr sent after the specified date and containing “Clinton” and at least one of “Lewinsky”, “Jones”, or “Flowers” will be ranked first.

5.2.2 Queries using negation

12. `Clinton ![Lewinsky Jones Flowers ]` Messages containing “Clinton” which do not include any of “Lewinsky”, “Jones”, or “Flowers” will be ranked first. The second tier of the ranking will include documents which mention “Clinton” and one or more of “Lewinsky”, “Jones”, or “Flowers”. The third and final tier would be made up of documents which contain none of the query terms. (In other words, they satisfy the second constraint but not the first.) Note that “Lewinsky”, “Jones”, and “Flowers” serve only as a constraint and do not contribute either positively or negatively to a document’s content relevance score. If `!` were replaced by `-` only the first tier would be presented.

13. `Clinton !f:Starr` Messages containing “Clinton” which do not include the word “Starr” in the from line will comprise the first tier of the ranking.

6 Case study 2: Intranet retrieval

An intranet search system which implements the design described above is in service at the Australian National University and may be tried out by anyone who is interested. Documentation and search facilities are accessible via [2]. Figure 2 lists the defined metadata classes.

A sample web query involving content and metadata constraints is shown in Figure 1 together with illustrative results.

The searchmail examples did not illustrate the treatment of cases where it cannot be determined whether or not a document satisfies a constraint. In the Web case, it is frequent for documents to be incompletely tagged with metadata, in which case the authorname or publisher may be unknown. Quite frequently, too, the “last modified” information may be missing.

```
a DC.Creator or Author
c DC.Coverage
d Last Modified date returned in http header
f DC.Format, DC.Type, DC.Content-type, DC.Context-type, Generator
h <a href> links
l DC.Language
m mailto
p DC.Publisher
r DC.Relation
s DC.Subject, DC.Description, keywords, description
t DC.Title or title
u hostname part of URL returned in http header
v filename part of URL returned in http header
x DC.Source
* anywhere
```

Figure 2: Metadata classes defined for the ANU intranet search system, together with the corresponding Dublin Core and Netscape metadata elements. The http header is the additional descriptive metadata supplied by the document server (http daemon) which supplied the page.

14. `a:Hawking aardvark` In this example, documents containing Hawking in the dc.creator field and aardvark in the document body will rank first. Documents containing aardvark with no dc.creator field will rank next, ahead of those in which there is a dc.creator field which does not contain Hawking.

7 Implementation details

The scheme described above has been implemented in a version (PADRE99) of the PADRE content-based document retrieval system [10] which uses a compressed inverted file index. This section describes the changes and additional structures necessary to implement the scheme.

7.1 Metadata constraints

Most of the metadata constraints supported by PADRE99 are determined by searching for a text term (word, phrase etc) within the appropriate metadata class. How this is done efficiently is described in Section 7.4.

7.2 Implementation of date constraints

During the lexical scanning phase the last-modified date supplied by the http daemon is parsed and interpreted as an 8-digit date of the form YYYYMMDD. An extra field in the document table is assigned to hold the date. A zero date indicates a parsing error or the absence of a last-modified date.

For efficient identification of all documents satisfying a particular date condition, an additional permutation array is envisaged but not yet implemented which allows location of a particular date using binary search.

7.3 Implementation of MAYBE values

As stated in Section 3.1, `maybe` values are independent of particular queries. A field in the document table records a bit map of the `maybe` state of all defined metadata classes. When indexing a document, the date is zeroed and all the maybe bits are set. Whenever an appropriate metadata element is encountered, the `maybe` bit for the corresponding metadata class is cleared.

7.4 PADRE index structures

PADRE content querying is based on a sorted term list accessed using binary search in which each entry includes an occurrence count and a pointer to the posting list for the term. The posting list contains a posting for every occurrence including the term position within the document. The postings are compressed using Elias gamma code [10]. Typically, the data structures necessary for query processing occupy about 25% of the size of the uncompressed raw text.

Support for phrases in queries is based on proximity relations and makes use of term position information.

To support the metadata/content model, each occurrence of a word in a metadata field is indexed as the word itself followed by a suffix to indicate the corresponding metadata class. Thus, occurrences of the word Craswell in a mail archive may generate dictionary entries for `craswell`, `craswell f`, `craswell t`, and so on, each with their own postings list.

This scheme ensures speed of query processing for metadata constraints and for content terms restricted to the body of the message. The penalty is an increase in the size of the term dictionary and a higher cost for unrestricted content searches. The latter require processing of the postings lists for all variants of a term. However, the use of suffixes starting with a blank ensures that the term dictionary entries are contiguous.

8 Evaluation

It is highly desirable to evaluate the efficiency and effectiveness of any new retrieval system. This has been done by analysing the performance of the production intranet search service at ANU.

The searchmail version of the system has also been implemented and is in private use by the authors and by an alpha tester. It appears to work satisfactorily but currently needs large table sizes to cope with email messages written in Hebrew and with binary attachments. More work is needed on the parser to enable the system to skip non-text attachments.

8.1 Effectiveness

There are as yet no test collections designed to test the type of combined metadata and content retrieval which we envisage. Agosti et al [1] report evaluations of metadata-based retrieval on Wall Street Journal and Library of Congress data sets, but their experiments were restricted to content description metadata such as keywords and titles. The queries they used did not apply relational constraints.

Extensive evaluation of the usefulness of the various basic features of this design (in the context of intranet search) is reported in [11]. The analysis is based on extensive analysis of query logs collected by the search service.

In summary, as far as could be ascertained from the query logs, the basic retrieval model described here appears to operate successfully in the intranet search environment but has not been fully exercised. This may be partly due to the fact that only relatively small pockets of the university intranet include significant amounts of metadata markup. It also seemed that searchers in this environment were not familiar with metadata querying, or were unaware that it was supported.

It was found that metadata constraints were used in about 11% of queries for the purpose of restricting searches to particular intranet hosts or directories. It is believed that most of these constraints are generated automatically by scripts used by local webmasters to implement departmental rather than full-organisation searches. Date constraints were applied in about 0.67% of queries but other metadata constraints, such as author, title and keywords, were made use of very infrequently[1]. This may be because they were not explicitly made visible in the graphical user interface.

Considering the quality of results returned, it was concluded that the basic content matching and constraint satisfaction mechanism described here would need to be augmented by heuristics and thesaurus-like mechanisms to ensure that the very short and very popular queries such as “library”, “scholarships”, “forestry” and so on, would efficiently guide the searcher to the correct home pages for those subjects, even when the relevant query words were not contained in those pages and when the query words were misspelled.

8.2 Efficiency

The ANU intranet search service was launched in July 1999 and runs on a relatively inexpensive PC system featuring dual 450 MHzPentium III processors (to minimise the effect of spidering and indexing on query processing) and 1024 MB of RAM. This configuration is grossly excessive for the current query load of around 450 queries per day, even though query-biased summaries of result documents are generated on the fly.

The architecture preserves the “scalable on scalable hardware” property described in [10].

The intranet is spidered each week and typically finds more than 175 distinct hosts, over 300,000 text/html pages and at least 3.5 gigabytes of text data. Indexing is completed in approximately 45 minutes elapsed time which is very much faster than the 12-15 hours taken for (polite) spidering.

Subjectively, response time of the system is very good. We plan to measure the maximum query processing rate of the complete system (including web interface, PADRE system, result page generator and document summariser). Hopefully these measurements will be available for the final version of this paper.

9 Discussion and conclusions

A search framework incorporating both content-matching and relational constraint satisfaction has been described. The framework is designed to provide useful and efficient search capabilities over document collections in which metadata is incompletely and perhaps inconsistently applied. It is not designed for complex structured metadata repositories.

The framework has been implemented for searching email archives and for web documents which make use of Netscape and/or Dublin Core metatags. An implementation is in production use at the Australian National University and has been the subject of an extensive evaluation (elsewhere) based on analysis of query logs. The system demonstrates very fast indexing and fast query processing on inexpensive commodity hardware.

Further work is needed to validate the metadata-content architecture in a metadata-rich, metadata-aware application environment. A suitable metadata plus content test collection is needed.

Neither of the applications thus far implemented have provided the opportunity to test the practical usefulness of the MAYBE state. In the ANU intranet metadata constraints are infrequently used. When they are used they are frequently made mandatory by the use of plus or minus operators, thereby eliminating consideration of MAYBE states. Metadata constraints are much more heavily used in mail archive search but the relevant metadata is almost universally present. A real digital library may provide a better framework for further investigations.

Another issue yet to be tested is whether the decision to allow metadata constraints to affect document content scores was justified. We will probably need to mount user trials to resolve this.

[1] We expect that metadata constraints will be used much more often in the searchmail application. This is both because sender and date metadata is always available and because these attributes are a very natural way of specifying messages to be retrieved.

References

[1] M. Agosti, F. Crivellari, and M. Melucci. The effectiveness of meta-data and other content descriptive data in web information retrieval. In *Proceedings of the Third IEEE Meta-Data Conference (META-DATA '99), Bethesda MD*, April 1999.

[2] Peter Bailey, David Hawking, and Francis Crimmins. P@NOPTIC home page. `www.panopticsearch.com`.

[3] Maria Luisa Barja, Tore Bratvold, Jussi Myllymaki, and Gabriele Sonnenberger. Informia: a mediator for integrated access to heterogeneous information sources. In *Proceedings of CIKM '98, Bethesda MD*, pages 234–241, New York, 1998. ACM Press.

[4] Samuel DeFazio, Amjad Daoud, Lisa Ann Smith, Jagannathan Srinivasan, Bruce Croft, and Jamie Callan. Integrating IR and RDBMS using cooperative indexing. In Fox et al. [6].

[5] Anne Fontaine. Sub-element indexing and probabilistic retrieval in the POSTGRES database system. Technical Report CSD-95-876, University of California at Berkeley, May 1995. `ftp://s2k-ftp.CS.Berkeley.EDU/pub/postgres/papers/`.

[6] Edward A. Fox, Peter Ingwersen, and Raya Fidel, editors. *Proceedings of SIGIR'95*, Seattle, Washington, July 1995. ACM Press, New York.

[7] Norbert Fuhr. Integration of probabilistic fact and text retrieval. In Nicholas Belkin, Peter Ingwersen, and Annelise Mark Pejtersen, editors, *Proceedings of SIGIR'92*, pages 211–222, Copenhagen, Denmark, June 1992.

[8] Norbert Fuhr. Probabilistic Datalog - a logic for powerful retrieval methods. In Fox et al. [6], pages 282–290.

[9] Norbert Fuhr and Thomas Rölleke. A probabilistic NF2 relational algebra for integrated information retrieval and database systems. *ACM Transactions on Information Systems*, 14(1):32–66, 1997.

[10] David Hawking. Scalable text retrieval for large digital libraries. In Carol Peters and Costatino Thanos, editors, *Proceedings of the First European Conference on Digital Libraries*, volume 1324 of *Lecture Notes in Computer Science*, pages 127–146, Pisa, Italy, September 1997. Springer, Berlin.

[11] David Hawking, Peter Bailey, and Nick Craswell. An intranet reality check for trec ad hoc. in preparation.

[12] David Hawking, Paul Thistlewaite, and Nick Craswell. ANU/ACSys TREC-6 experiments. In Ellen Voorhees and Donna Harman, editors, *Proceedings of the TREC-6 Connference*, pages 275–290, Gaithersburg, MD, 1997. NIST. `trec.nist.gov/pubs/trec6/papers/anu.ps.gz`.

[13] National Institute of Standards and Technology. TREC home page. `trec.nist.gov/`, 1997.

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

[15] Thomas Rölleke and Norbert Fuhr. Composite documents and uncertain inference. In *Proceedings of the 2nd Workshop on IR, Uncertainty and Logic*, Glasgow University, 1996.

[16] Thomas Rölleke and Norbert Fuhr. Querying for facts and content in hypermedia documents. In *Proceedings of the 2nd Workshop on IR, Uncertainty and Logic*, Glasgow University, 1996.

[17] Thomas Rölleke and Norbert Fuhr. Retrieval of complex objects using a four-valued logic. In Hans-Peter Frei, Donna Harman, Peter Schäuble, and Ross Wilkinson, editors, *Proceedings of SIGIR'96*, pages 206–214, Zurich, Switzerland, August 1996. ACM Press, New York.

[18] E. M. Voorhees and D. K. Harman, editors. *Proceedings of TREC-7*, Gaithersburg MD, November 1998. NIST special publication 500-242, `trec.nist.gov`.