A Comparison of Techniques for Selecting Text Collections

Daryl J. D'Souza James A. Thom Justin Zobel
Department of Computer Science, RMIT University
GPO Box 2476V, Melbourne 3001, Australia
$\{$*djds, jat, jz*$\}$*@cs.rmit.edu.au*

Abstract

Techniques for evaluating queries against a distributed text document database allow uniform access to separate collections in the database. One such technique is to first choose a subset of collections, via a selection index. The index captures information about each collection such as terms occurring in documents, term statistics, and collection statistics. A possible implementation of such an index is a lexicon, which maintains a complete list of terms in the database. Another approach is to partially index the database by extracting fewer terms but maintaining some information about each document. In this paper we explore three collection-ranking techniques, two based on lexicons and the other based on partial document indexes. Our experiments show that in most cases the lexicon approaches outperform the partial index approach.

1. Introduction

In text searching tasks, such as locating information on the Web, it is attractive to be able to present a query simultaneously and transparently to a large set of text collections. However, it is costly to evaluate each query at every collection. Collection ranking potentially offers significant benefits for a distributed document system that supports query-based retrieval of documents from a set of document collections, that is, from a distributed document database.

Ranking the collections prior to exhaustive interrogation of the individual sets of documents provides an opportunity to evaluate each query on fewer databases. Collection ranking can also be used directly. For searching the world wide web, for example, most current search engines return lists of matching documents. However, a search engine could instead return site addresses of document collections ranked in the order of relevance to the query. The user could then select from among the top-ranked collections for searching for individual documents.

There are two broad approaches to collection ranking. One is to gather the set of distinct terms, or *lexicon*, from each collection, together with in-collection and cross-collection statistics. These can then be used to estimate the appropriateness of each collection to a query. The other approach is to index every document in each collection, then select at query time the collection with the greatest number of or the most promising documents. Fully indexing each document is expensive—it requires holding some information about every term in each document. But partial index information need take no more space than that required by a lexicon index, as shown by D'Souza and Thom [2]; although partial information does not yield good ranking of documents, it may allow good ranking of collections.

We explore several ranking techniques and compare them in a consistent test environment. We propose several variants of an *n-term indexing* scheme [2] based on partial information about each document, and compare its effectiveness to that of two previously published schemes based on the lexicon approach. Our experiments with two test collections show that the lexicon approach is superior.

The remainder of this paper is organised as follows. In Section 2 we discuss related work and explain collection selection and ranking. Our motivation for the choice of ranking algorithms for exploration and the details of the algorithms is presented in Section 3. Experiments and results are in Section 4, and in Section 5 we report our conclusions and plans for further work.

2. Background

The problem of identifying or selecting candidate collections for further optional interrogation by a user is known as the *collection selection* problem, and has been widely investigated [1, 2, 4, 5, 6, 8, 9, 11, 12, 13]. A collection is ranked according to its numerical *goodness* score [5, 6, 12], computed from the information available about collections in a *collection directory* or index. The *goodness* score, denoted $G(q, c)$, is an estimate of the relevance of a collection c to a query q. Collections are then selected according to their *goodness* score. Such candidate collections are subsequently interrogated with the given query, and matching documents retrieved. A problem related to collection selection is that posed by meta-search engines, which must choose appropriate search engines for evaluation of queries.

The aim of a distributed document system is to be as effective as a centralised system with the same document set that the former serves. An obvious solution is to fully index every document in every collection, but this solution may be infeasible for large databases, because of the cost of holding the index information and initially gathering it.

Another solution is to generate *lexicons* from the collections; a lexicon is a complete list of terms in a collection and some associated statistics, such as document frequency. Various lexicon approaches have been reported in the literature [1, 4, 5, 6, 8, 9, 11, 12, 13].

In an approach proposed by D'Souza and Thom [2], only a subset of terms from each document is indexed. The technique reported in this initial work used the first n terms from each document, where n can be varied; the rationale for this is that important terms from the title, abstract, and first paragraph may suffice for ranking. Increasing n improves the quality of document ranking, at the cost of increasing the cost of representing the index. These results showed that n-term indexes are a poor basis for document ranking. Moreover, in subsequent experiments we have observed little improvement with more sophisticated term selection techniques, such as choosing terms based on their frequency in the document and rareness overall. One hypothesis tested in this paper is whether n-term indexes can help select collections.

3. Collection ranking

We describe in this section the ranking algorithms chosen for our comparison. Our choice was motivated by several factors. Traditionally lexicon approaches have been investigated and we were interested to compare several such methods with our partial-index approach, reported in [2]. The lexicon approaches independently investigated by Zobel [13] and Yuwono and Lee [12] showed that relevance-based ranking by lexicons is effective, and in the latter case superior to some lexicon approaches mentioned in Section 2. We were interested in comparing them with the n-term method within a consistent test environment consisting of large databases of document collections, short and long queries, and where relevance judgements were available.

3.1. Lexicon inspection

A *lexicon* is a set of terms that occur in a document collection. A lexicon index contains the terms occurring in every collection, and, for each term, a list identifying which collections hold the term and what the statistics associated with the term are. In effect, each collection is indexed as if it were a single document. One exploration of the lexicon approach was conducted by Zobel [13] in his Lexicon Inspection (*LI*) system. LI maintains an index in which every unique term across the database of collections is represented. Each index entry contains a term and a list of corresponding collections in which the term appears; accompanying each entry in the list is a term statistic, where the choice of statistic depends on the collection ranking formulation employed. The best ranking measure identified in these experiments was based on the inner product (I) measure of document similarity:

$$G(q,c)_I^{LI} = \sum_{t \in q \& c} w_{q,t} \cdot w_{c,t}$$

where the rank denoted by $G(q,c)_I^{LI}$ for a given query q and collection c is the sum of products of query-term and collection-term weights. Each weight $w_{x,t}$ (where x is query q or collection c) is defined as:

$$w_{x,t} = w_t \cdot log(f_{x,t} + 1)$$

where w_t represents the importance of term t across all collections and is given by:

$$w_t = log(N/f_t + 1)$$

where N is the size of the database (number of document collections) and f_t is the number of documents in the database that contain t. Finally, $f_{x,t}$, which is either $f_{q,t}$ or $f_{c,t}$ is the query term frequency or the document frequency in collection, respectively, for term t.

3.2. D-WISE

Yuwono and Lee [12] investigated a lexicon approach as part of the *D-WISE* research project (Distributed World Wide Web Index Servers and Search Engine). D-WISE employs a measure called the *Cue-Validity Variance* or *CVV*; it measures the usefulness of the term t for distinguishing one collection from another and the larger the measure, the more useful the term. Under this scheme $G(q,c)_{CVV}^{DWISE}$ for a given query and collection is given by:

$$G(q,c)_{CVV}^{DWISE} = \sum_{t \in q} CVV_t \cdot f_{c,t}$$

where $f_{c,t}$ is the collection frequency of term t in collection c, and the Cue-Validity Variance is:

$$CVV_t = \frac{\sum_{c=1}^{N}(CV_{c,t} - CV_t)^2}{N}$$

The computation of CVV_t is based on the *cue validity* or $CV_{c,t}$ which measures the degree to which t distinguishes documents in collection c from those in the other collections, and is defined thus:

$$CV_{c,t} = \frac{\frac{f_{c,t}}{N_c}}{\frac{f_{c,t}}{N_c} + \frac{\sum_{k=1 \wedge k \neq c}^{N} f_{k,t}}{\sum_{k=1 \wedge k \neq c}^{N} N_k}}$$

where N_c is the size (number of documents) in collection c. Finally, CV_t is the average cue-validity of term t over all the collections in the database, and is given by:

$$CV_t = \frac{\sum_{c=1}^{N} CV_{c,t}}{N}$$

3.3. n-term indexing

The *n-term indexing* (*NTI*) scheme was first reported in [2] and represents a compromise between the lexicon approaches and full-text indexing. NTI is a partial-index scheme that indexes relatively fewer terms from each collection in the database, thereby permitting document and collection references to be maintained. Thus each index entry consists of a unique term and a list identifying the containing documents and collections, and a statistic as per lexicon schemes.

There are several ways to select the terms for indexing in NTI, leading to a range of algorithms. We limit our discussion to the *first-n* variant which indexes the first n unique terms from each document. The choice of n itself may vary; the choice of $n = 30$ was based on an initial exploration of storage costs and found that such a value for n rendered an n-term index approximately comparable in size to a corresponding LI index, for the same database. Thus, for $n = 30$, construction of an NTI index proceeds by selecting from each document the first 30 unique terms. In the experiments reported here we used this value for n.

As for LI several ranking protocols are possible within NTI. Our approaches exploit the fact the NTI retains document references, beyond the single statistic maintained in the lexicon schemes. Analytical formulations for these ranking algorithms are as follows (a brief description precedes each formulation):

1. Rank of highest ranked document.

$$G(q,c)^{NTI}_{naive} = \max_{d \in C_c} sim(q,d)$$

2. Sum of document ranks.

$$G(q,c)^{NTI}_{sumsim} = \sum_{d \in C_c} sim(q,d)$$

3. Inverse document ordinal (in rank list).

$$G(q,c)^{NTI}_{invrank} = \sum_{d \in C_c} \frac{1}{r_d + K}$$

4. Sum of document ranks squared.

$$G(q,c)^{NTI}_{sumsimsqr} = \sum_{d \in C_c} sim(q,d)^2$$

5. Rank divided by document ordinal in rank list.

$$G(q,c)^{NTI}_{simdivrank} = \sum_{d \in C_c} \frac{sim(q,d)}{r_d}$$

where C_c is the set of documents ranked and which appear in collection c, $sim(q,d)$ is the similarity of document d and query q, r_d is the rank ordinal of document d, and K is a constant, arbitrarily set to 10.

4. Experiment

We present in this section the essential process and data ingredients necessary to evaluate the foregoing ranking algorithms. We first provide information about database and query profiles, and then present our evaluation and results.

4.1. Database and query profiles

We used the two data and query sets used by Zobel to evaluate the three selection methods: disk2 with queries 51 to 150; and disk3 with queries 202 to 250, of the TREC [7] corpus. These data sets were split into several partitions each representing a separate database collection of documents.

The disk2 set was split into 43 collections varying from 1,600 to 7,500 documents each. Each such collection was created from a single source, such as one month of Wall Street Journal articles. The queries averaged over 100 terms each. Relevance judgements for this data and query set numbered around 11,000.

The disk3 set was distributed across 91 collections by randomly selecting split points for each new collection. The collection sizes varied from 14 to 23,000 documents (approximately averaging 5,000). The query set (202-250) averaged 10 terms, representing short queries and there were around 3,300 relevance judgements for this data and query set.

In both cases document and query terms were case-folded and stemmed before constructing indexes and querying them, respectively. Additionally, queries were stopped. In the case of the NTI method we used the `mg` system [10] to construct the index. LI and DWISE indexes were generated from the data sources directly.

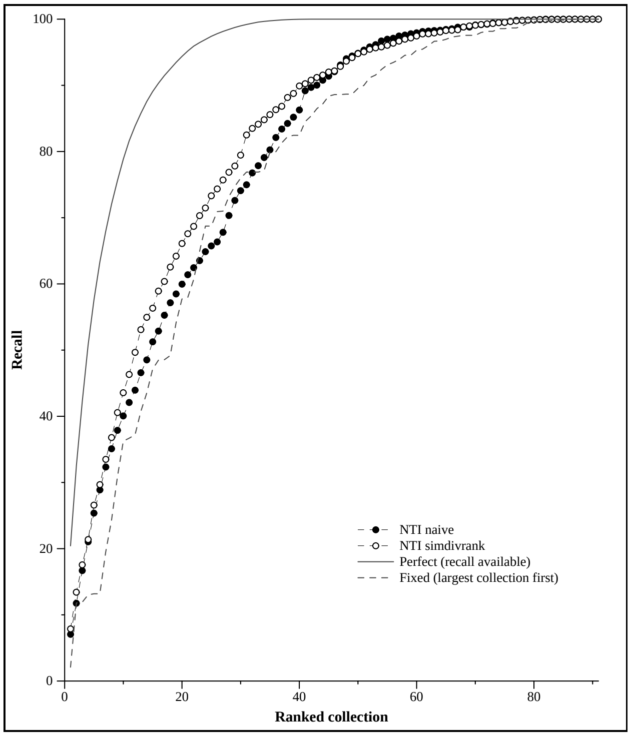

Figure 1. *Performance of NTI (naive, simdivrank) rankings for TREC disk3.*

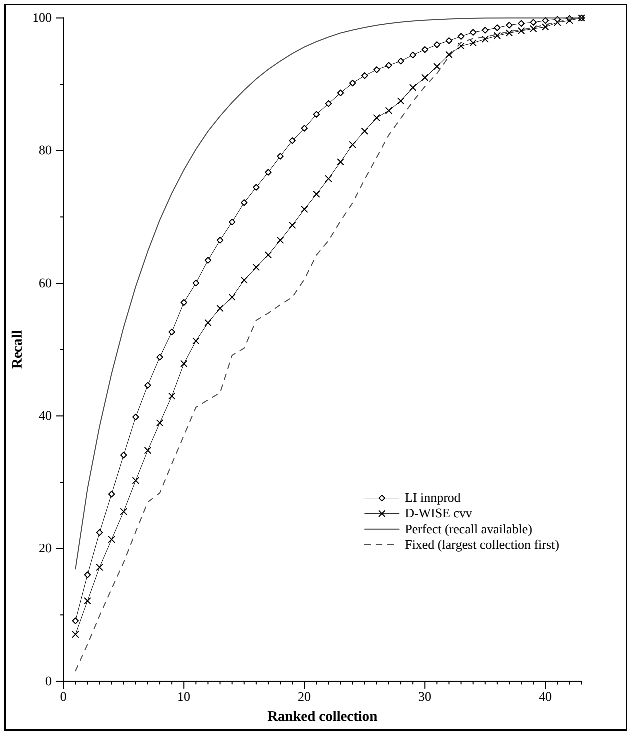

Figure 2. *Performance of LI (innprod) and D-WISE (cvv) rankings for TREC disk2.*

4.2. Evaluation and results

LI was evaluated by Zobel [13] for several ranking formulations (including the afore-mentioned inner product formulation) and measured by two yardsticks. We chose the relevance based evaluation which measured the ability of a scheme to identify collections with known relevant documents. As well the same baselines were employed: *perfect* and *fixed*. The *perfect* baseline ordered collections according to the most number of relevant documents while the *fixed* baseline ordered them from largest to smallest based on the number of documents. Other evaluations are explored by Zobel [13] and Yuwono et al. [12].

We first explored the NTI (*first-n* term selection algorithm) ranking algorithms described earlier for the two data sets. Figure 1 presents results for two of the ranking algorithms (*naive* and *simdivrank*) for disk3. The method *simdivrank* outperformed the *naive* and the other methods described in Section 3.3.

A comparison of the lexicon performances for LI and D-WISE is graphically presented in Figure 2 (disk2) and for LI, D-WISE and NTI (*simdivrank*) is graphically presented in Figure 3 (disk3). The graphs show that LI outperforms both D-WISE and NTI (*simdivrank*), and the performance of D-WISE and NTI (*simdivrank*) is similar.

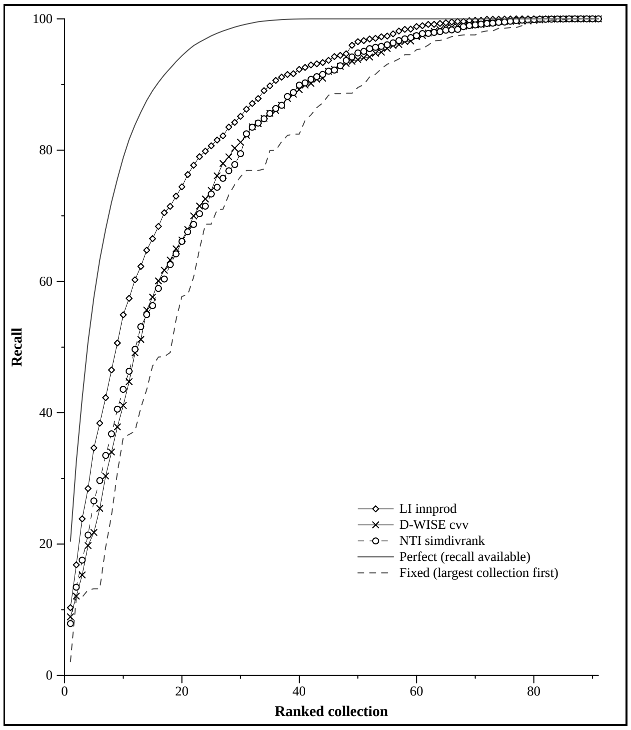

Figure 3. *Performance of LI (innprod), D-WISE (cvv) and NTI (simdivrank) ranking for TREC disk3.*

A more extensive set of results for LI, D-WISE and NTI (for the ranking formulations presented in Section 3.3, and a range of term selection algorithms) and for disk2 and disk3, is presented by D'Souza et al. [3]. The results for the NTI *first-n* algorithm, and an improved variation thereof, are consistent with those presented in this paper.

5. Conclusions and Further Work

In this paper we presented three collection selection indexing schemes and ranking algorithms based on each scheme. Two of these schemes used a lexicon; the other employed a partial index. We evaluated the ranking algorithms for these schemes using a relevance based ranking evaluation and for two data and query sets from the TREC corpus. The resulting collection ranking performance of the lexicon inspection (LI) approach exceeded that of the other approaches.

We are presently exploring the efficacy of the NTI scheme for collection *fusion*, which is the process of merging individual document sets returned by queried collections. NTI schemes maintain document identifiers in their indexes presenting an opportunity to directly rank the documents from a distributed set of collections, thereby obviating any need to first select and rank the collections, as is required in the lexicon approach.

Acknowledgements

This work was supported by the Multimedia Database Systems group at RMIT University and the Australian Research Council.

References

[1] J. P. Callan, Z. Lu, and W. B. Croft. Searching distributed collections with inference networks. In E. A. Fox, P. Ingwersen, and R. Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–28. ACM, July 1995.

[2] D. D'Souza and J. Thom. Collection selection using n-term indexing. In *Proceedings of the Second International Symposium on Cooperative Database Systems for Advanced Applications (CODAS'99)*, Wollongong, NSW, Australia, March 1999. To appear in Springer-Verlag (Singapore) publication.

[3] D. D'Souza, J. Thom, and J. Zobel. A comparison of techniques for selecting text collections. Technical Report TR-99-9, Department of Computer Science, RMIT University, Melbourne, VIC 3001, Australia, 1999. Abstract: http://www.cs.rmit.edu.au/reports/1999/99-9.html.

[4] J. C. French, A. L. Powell, C. L. Viles, T. Emmitt, and K. J. Prey. Evaluating database selection techniques: A testbed and experiment. In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 121–129. ACM, August 1998.

[5] L. Gravano and H. Garcia-Molína. Generalising GlOSS to vector-space databases and broker hierarchies. In *Proceedings of 21st International Conference on Very Large Data Bases, September 11-15, Zurich, Switzerland*, September 1995.

[6] L. Gravano, H. Garcia-Molína, and A. Tomasic. The effectiveness of GlOSS for the text database discovery problem. In *Proceedings of SIGMOD 94*, pages 126–137. ACM, May 1994.

[7] D. K. Harman. Overview of the first text retrieval conference. In D. Harman, editor, *Proceedings of Text Retrieval Conference*, pages 1–20, Washington, November 1992. National Institute of Standards and Technology Special Publication 500-207.

[8] A. Moffat and J. Zobel. Information retrieval systems for large document collections. In D. Harman, editor, *Proceedings of Third Text Retrieval Conference (TREC-3)*, pages 85–93, Washington, 1994. National Institute of Standards and Technology Special Publication 500-225.

[9] E. M. Voorhees. Siemens TREC-4 report: Further experiments with database merging. In D. Harman, editor, *Proceedings of Fourth Text Retrieval Conference (TREC-4)*, Washington, October 1996. National Institute of Standards and Technology Special Publication 500-225.

[10] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and indexing documents and images*. Van Nostrand Reinhold, New York, 1994.

[11] J. Xu and J. P. Callan. Effective retrieval with distributed collections. In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 112–120. ACM, August 1998.

[12] B. Yuwono and D. L. Lee. Server ranking for distributed text retrieval systems on the internet. In *Proceedings of the 5th International Conference on Database Systems for Advanced Applications (DASFAA'97)*, pages 41–49, Melbourne, Victoria, Australia, April 1997.

[13] J. Zobel. Collection selection via lexicon inspection. In *Proceedings of the 2nd Australian Document Computing Symposium (ADCS'97), April 1997*, pages 74–80, Melbourne, Victoria, Australia, April 1997.