UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

UIPATH, INC.,

Petitioner,

v.

RULE 14 LLC.,

Patent Owner.

Case No. IPR2024-01306
U.S. Patent No. 11,048,712

DECLARATION OF CHELSEA A. STEADMAN, ESQ.

I, Chelsea A. Steadman, make the following Declaration pursuant to 28 U.S.C. § 1746:

1. I am an associate with the law firm of Volpe Koenig, attorneys for Petitioner in this matter.

2. According to the Science Direct website page https://www.sciencedirect.com/science/article/abs/pii/S0306457303000086, the publisher of the publication D'Souza et al., Collection Selection for Managed Distributed Document Databases, Information Processing & Management, Volume 40, Issue 3 (May, 2004) ("D'Souza Paper"), is Elsevier. A true and correct copy of a screen capture of the website page found at the URL https://www.sciencedirect.com/science/article/abs/pii/S0306457303000086 that I created on August 15, 2024 is attached as Exhibit A.

3. On May 7, 2024, I sent an email to JournalCustomerService-usa@elsevier.com requesting information on obtaining a copy of the D'Souza Paper.

4. On May 8, 2024, I corresponded and spoke with Erica Smith, an Elsevier Account Manager, regarding obtaining a copy of the D'Souza Paper.

5. On May 8, 2024, Ms. Smith e-mailed a copy of the D'Souza Paper to me, a true and correct copy of which is attached hereto at Exhibit B. A true and correct copy of the D'Souza Paper is also included as Ex. 1010 in connection with the IPR Petition filed in this matter.

6. I hereby declare under penalty of perjury under the laws of the United States that all statements made herein of my knowledge are true, and that all statements made on information and belief are believed to be true. I understand willful false statements and the like are punishable by fine or imprisonment, or both, under § 1001 of Title 18 of the United States Code.

Dated: 8/16/2024 By: _____
Chelsea A. Steadman, Esq.

EXHIBIT A

Captured at: 2024/08/15 01:31 PM URL: https://www.sciencedirect.com/science/article/abs/pii/S0306457303000086

Access through your institution Purchase PDF

Information Processing & Management
Volume 40, Issue 1, May 2004, Pages 127-146

Collection selection for managed distributed document databases

Darryl D'Souza, James A. Thom, Justin Zobel

Show more

Add to Mendeley Share Cite

https://doi.org/10.1016/S0306-4573(03)00064-4 Get rights and content

Abstract

In a distributed document database system, a query is processed by passing it to a set of individual collections and collating the responses. For a system with many such collections, it is attractive to first identify a small subset of collections as likely to hold documents of interest before interrogating only this small subset in more detail. A method for choosing collections that has been widely investigated is the use of a selection index, which captures broad information about each collection and its documents. In this paper, we re-evaluate several techniques for collection selection.

We have constructed new sets of test data that reflect one way in which distributed collections would be used in practice, in contrast to the more artificial division into collections reported in much previous work. Using these managed collections, collection ranking based on document surrogates is more effective than techniques such as CORI that are based on collection lexicons. Moreover, these experiments demonstrate that conclusions drawn from artificial collections are of questionable validity.

Introduction

In meta-search, a distributed document database consists of a set of document collections, where each collection may be held at a different location. Typically, such a database is searched via presentation of a ranked query by a meta-search broker, which (in principle) broadcasts the query to the document collections, collates their responses, and presents a consolidated list of answers to the user. Given sufficient information about document and term statistics, such a meta-search engine can be as effective as a system (Meng, Yu, & Liu, 2002) in which all the documents are held monolithically in a single collection. However there is a trade-off between competing goals: minimising the amount of information that needs to be centralised, and maximising the retrieval effectiveness.

Standard monolithic search engines have proved their ability to effectively resolve users' queries in contexts such as the web, where large numbers of documents of interest can be gathered by crawling, and the costs of maintaining a large central index are amortised over large number of queries. In other contexts, meta-search (Meng et al., 2002) is clearly the more attractive option. For example, some individual collections, such as online encyclopaedias, may not be browsable. A meta-search client can be relatively lightweight, and thus can easily be replicated on a number of machines or supported on a cheap platform. Meta-search is also of value on intranets. In typical companies, each individual personally manages on their desktop machine a collection of documents related to their work, which they have either authored or archived. If such collections are globally searchable across the company, meta-search is an effective mechanism for accessing them.

Broadcasting queries to all collections could well be costly or impractical, thus it is attractive to first rank the collections in the database, in decreasing order of a metric that measures how useful a collection is likely to be for that query. The user can then query the top-ranked collections directly, or the meta-search engine can do so without user intervention. Ranking the collections then provides a form of collection selection.

Several techniques have been proposed for collection selection. There are two general kinds of technique based on indexing. One is to gather the set of distinct terms, or lexicon, from each collection, together with some inter- and intra-collection statistics such as the number of documents containing each term; these statistics may then be used to estimate the appropriateness of each collection to a query. For large collections, such a lexicon is typically around 1% of the size of the documents it describes, but relative size increases as database size falls. Since the bulk of the space is the distinct words, and since many words are repeated between collections, as a rough guide a selection index is no larger than the space required for the distinct words. The well-known CORI (Callan, Lu, & Croft, 1995) method is based on lexicons.

The other kind of selection technique is to index documents in the database of collections, then select at query time the collection with the most promising documents. Fully indexing each document is expensive—and in effect yields a monolithic implementation—so some compromise is required. One is to select representative documents from each collection, a method that is implicit in clustering and centroids (Salton & McGill, 1983), which to our knowledge has not been evaluated. Another is to index a surrogate representation, or summary, of each document. Standard document summaries are produced for human consumption and are not an ideal surrogate for this task; they include stop-words, for example. A surrogate for indexing purposes can be created by methods such as selecting the k highest-weight terms from the document, a method also used for choosing terms for query expansion. We have explored such strategies in previous work, but did not find them superior to using lexicon-based methods.

These methods assume that the systems cooperate, an assumption that is made throughout this paper. Other approaches to distributed retrieval are closer to the model used in Web meta-search systems. In principle, particularly in the absence of cooperation, it would be possible for a collection to contain many relevant documents, be highly ranked, but for the collection's ranking mechanism to be unable to find the documents. In practice, in cooperative systems in which information such as term statistics are shared, this is no more likely than in a monolithic system (de Kretser, Moffat, Shimmin, & Zobel, 1998). Thus it is reasonable to compare systems by their ability to find collections with relevant documents, while noting that there remains the issue of how to find documents within collections and combine these results.

A shortcoming of some previous research in this area is that the comparison of selection methods has been on test collections that do not represent likely patterns of distribution in real multi-collection environments. In some of the TREC-based evaluations, for example, the collections were created by division into large sets of similar size (Callan et al., 1995; Voorhees, 1996; Voorhees, Gupta, & Johnson-Laird, 1995). More seriously, some comparison has been on collections in which a monolithic collection has been divided in a way that is somewhat artificial, where the allocation of documents to collections has either been random, or based on chronology: with the TREC newspaper collections, for example, one common subdivision has been month by month. These collections are unlikely to show a great deal of topic specificity, other than the fraction of the articles that follow a range of threads of topical interest. For meta-search on the web, where each collection is a web site, such a division is highly improbable, as is evidenced by the TREC-8 Small Web Track data source (Hawking, Voorhees, Craswell, & Bailey, 2000) used by Craswell, Bailey, and Hawking (2000) in their evaluation of web-based server selection.

Another way of dividing documents into the test collections is by the way they are used or created. Categorisation techniques can be used to divide documents among a limited number of areas but such an approach assumes that the documents are created centrally, then distributed. Furthermore, to date there has been no application of categorisation to the collection selection problem (Sebastiani, 2002). An alternative is to use attributes of the documents that reflect their origin: for example, their authors, or the place they were written, or some other simple organisational protocol. We believe such an organisation of collections more closely reflects actual workplace practices and provides a valuable contrast to other test collections. We call a group of organised collections a managed distributed document database because documents are assigned to collections in some more-or-less systematic fashion. Examples include databases where the collections are individual web sites; databases where each collection is the set of documents created at a particular office (such as a branch of a government department) or by a particular person (such as a journalist); or databases where each collection is drawn from a particular division of an organisation. To our knowledge only one previous paper, by Larkey, Connell, and Callan (2000), explicitly explores managed collections; unfortunately, as discussed later, their results are flawed by a serious methodological error. While some of the earlier work with dividing the TREC data into collections, such as that of Voorhees et al. (1995), can be described as managed, the few collections used and the fact that they were created to be of the same size means that they are unlikely to be representative of real data.

In this paper we re-evaluate a range of collection selection techniques on two sets of collections, where the documents are distributed into around 500 and 2000 collections respectively. The documents are distributed both by methods used previously (randomly and by chronological ordering) and by a document attribute reflecting their origin (such as document author). These experiments produce two major results. One is that distribution by attribute allows more effectively selection of collection, or, alternatively, is better at placing relevant documents together, despite the simplicity of the distribution method and the large number of collections involved. The other is that, on a managed collection, selection by document surrogate is clearly superior to selection by lexicon methods.

Our experiments also pose questions about previous results in the area. With a chronological division into collections, the differences between the various methods are small, whether the queries are short or long. Indeed, on these collections many of the methods we tested do little better than the fixed strategy of choosing the largest collection first. When the collections are all of similar size (another unrealistic aspect of many earlier experiments), differences in performance are even less conspicuous. In our experiments, most of the differences in performance observed on these collections are not statistically significant. A further observation concerns long queries, which, while not widely used in some contexts (Spink, Wolfram, Jansen, & Saracevic, 2001), are for example generated by query expansion and moreover provide an interesting point of comparison. Not only are long queries better at collection selection—which in itself is unsurprising—but reveal even greater differences between the methods, and thus provide a way of discriminating between methods.

A particular result is that our experiments show that for managed data CORI is markedly inferior to the other methods tested. To the best of our knowledge previous work has not explored a comparative study of CORI and the lexicon methods presented in this paper. However, since CORI has been widely reported as an effective collection selection method, its poor performance in these experiments is deeply surprising.

Access through your organization
Check access to the full text by signing in through your organization.
Access through your institution

Section snippets

Collection selection

The collection selection problem has been widely investigated. In most of the techniques that have been proposed, each collection is ranked according to a goodness score, computed from a selection index. The score is a measure of how likely collection c is to contain documents that are relevant to query q. Collections are ranked according to their goodness score, and the selected (candidate) collections are subsequently interrogated with the given query, and the matching documents retrieved....

Evaluation of collection selection algorithms

In evaluating the performance of collection selection algorithms, the question that needs to be addressed is: How good is an algorithm at ranking collections such that the number of relevant documents returned is maximised? The question of evaluation is a choice between selecting collections that contain highly similar documents or those that contain highly relevant documents (Zobel, 1997). If the aim is to emulate a retrieval mechanism, then the former choice is appropriate; this was the ...

Experiments

To test our hypothesis that collection selection is influenced by the way documents are organised within a distributed document database, we ran experiments on six distinct databases, each derived from the same original set of documents. All documents were sourced from the Associated Press TREC volumes 1 and 2 (Harman, 1995). All these documents are in SGML and contain at least a DOCNO field, and optionally one or more TEXT fields and HEADLINE fields. A (heavily edited) portion of such a ...

Conclusions

We have compared a range of index-based collection selection techniques on several databases. These experiments contrast the well-known CORI lexicon-based method with other lexicon-based and surrogate-based methods. They also contrast both the performance available and the changes in behaviour observed for different approaches to constructing test collections.

Our results are unequivocal. Of the methods tested, CORI is clearly the weakest, coming last or near-last in all 12 comparisons. The best ...

Acknowledgements

This work used computing facilities supported by the Australian Research Council. We would also like to thank the anonymous referees, whose comments we found helpful in revising this paper....

References (28)

D. Harman
Overview of the second text retrieval conference (TREC-2)
Information Processing and Management, [1995]

Broglio, J., Callan, J., Croft, W. B., & Nachbar, D. [1994]. Document retrieval and routing using the INQUERY system....

J.P. Callan
Distributed information retrieval

Callan, J., Connell, M., & Du, A. [1999]. Automatic discovery of language models for text databases. In Proceedings of...

Callan, J. P., Lu, Z., & Croft, W. B. [1995]. Searching distributed collections with inference networks. In E. A. Fox,...

Callan, J., Powell, A. L., French, J. C., & Connell, M. (2000). The effects of query based sampling in automatic...

Craswell, N., Bailey, P., & Hawking, D. (2000). Server selection on the world wide web. In Proceedings of the fifth...

de Kretser, O., Moffat, A., Shimmin, T., & Zobel, J. [1998]. Methodologies for distributed information retrieval. In M....

D'Souza, D., & Thom, J. A. [1999]. Collection selection using n-term indexing. In Y. Zhang, M. Rusinkiewicz, & Y....

D'Souza, D., Thom, J. A., & Zobel, J. (2000). A comparison of techniques for selecting text collections. In Proceedings...

View more references

Cited by (18)

Knowledge based collection selection for distributed information retrieval
2010, Information Processing and Management

Citation Excerpt :
...The dataset is used by several tracks of the TREC conference. If the dataset is divided randomly or have on chronology, the performance of collection selection methods would be affected and experimental results would be of questionable validity (D'Souza, Thom, & Zobel, 2004; Kulkarni & Callan, 2010). Therefore, we divide this dataset into 100 collections by applying topical clustering algorithm (Kulkarni & Callan, 2010)....

Show abstract

Using query logs to establish vocabularies in distributed information retrieval
2007, Information Processing and Management

Citation Excerpt :
...Although this approach is more effective than Carmel's method in most cases, the loss in average precision compared to a full-text baseline is significant. D'Souza, Thom, and Zobel (2004) discuss surrogate methods for pruning, where only the most significant words are kept for each document. In this approach, the representation set is not the collection's vocabulary but is instead a complete index for the surrogates....

Show abstract

NeuralSearchX: Serving a Multi-billion-parameter Reranker for Multilingual Metasearch at a Low Cost »
2023, CEUR Workshop Proceedings

NeuralSearchX: Serving a Multi-billion-parameter Reranker for Multilingual Metasearch at a Low Cost »
2022, arXiv

Embedding based learning for collection selection in federated search »
2020, Data Technologies and Applications

Web data source selection technologies »
2013, Ruan Jian Xue Bao/Journal of Software

View all citing articles on Scopus

View full text

Copyright © 2003 Elsevier Ltd. All rights reserved.

Recommended articles

Employing fisher discriminant analysis for Arabic text classification
Computers & Electrical Engineering, Volume 66, 2018, ... Dia AbuZeina, Fouad S. Al-Anzi

Photodegradation of formylmethylflavin by sidechain and isoalloxazine ring cleavage...
Journal of Photochemistry and Photobiology A: Chemi... Iqbal Ahmad, ... Sodio Hefeez Kazi

Location-aware service applied to mobile short message advertising: Design,...
Information Processing & Management, Volume 51, 5... Chaomeng Wu, ... Stan Huang

Show 3 more articles

Article Metrics

Citations

Citation Indexes: 17

Captures

Readers: 14

PLUMX View details

EXHIBIT B

ELSEVIER

Information Processing and Management 40 (2004) 527–546

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

Collection selection for managed distributed document databases

Daryl D'Souza *, James A. Thom, Justin Zobel

School of Computer Science and Information Technology, RMIT University, Melbourne, Vic. 3001, Australia

Received 16 September 2002; accepted 31 January 2003

Abstract

In a distributed document database system, a query is processed by passing it to a set of individual collections and collating the responses. For a system with many such collections, it is attractive to first identify a small subset of collections as likely to hold documents of interest before interrogating only this small subset in more detail. A method for choosing collections that has been widely investigated is the use of a selection index, which captures broad information about each collection and its documents. In this paper, we re-evaluate several techniques for collection selection.

We have constructed new sets of test data that reflect one way in which distributed collections would be used in practice, in contrast to the more artificial division into collections reported in much previous work. Using these *managed* collections, collection ranking based on document surrogates is more effective than techniques such as CORI that are based on collection lexicons. Moreover, these experiments demonstrate that conclusions drawn from artificial collections are of questionable validity.
© 2003 Elsevier Ltd. All rights reserved.

Keywords: Distributed document database; Collection selection; Meta-indexing; CORI

1. Introduction

In meta-search, a distributed document database consists of a set of document collections, where each collection may be held at a different location. Typically, such a database is searched via presentation of a ranked query to a meta-search client, which (in principle) broadcasts the query to the document collections, collates their responses, and presents a consolidated list

* Corresponding author. Tel.: +613-9925-2927; fax: +613-9962-1617.
E-mail addresses: djds@cs.rmit.edu.au (D. D'Souza), jat@cs.rmit.edu.au (J.A. Thom), jz@cs.rmit.edu.au (J. Zobel).

0306-4573/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0306-4573(03)00008-6

IPR2024-01306
UIPATH EXHIBIT 1026

of answers to the user. Given sufficient information about document and term statistics, such a meta-search engine can be as effective as a system (Meng, Yu, & Liu, 2002) in which all the documents are held monolithically in a single collection. However there is a trade-off between competing goals: minimising the amount of information that needs to be centralised, and maximising the retrieval effectiveness.

Standard monolithic search engines have proved their ability to effectively resolve users' queries in contexts such as the web, where large numbers of documents of interest can be gathered by crawling, and the costs of maintaining a large central index are amortised over large number of queries. In other contexts, meta-search (Meng et al., 2002) is clearly the more attractive option. For example, some individual collections, such as online encyclopaedias, may not be browsable. A meta-search client can be relatively lightweight, and thus can easily be replicated on a number of machines or supported on a cheap platform. Meta-search is also of value on intranets. In typical companies, each individual personally manages on their desktop machine a collection of documents related to their work, which they have either authored or archived. If such collections are globally searchable across the company, meta-search is an effective mechanism for accessing them.

Broadcasting queries to all collections could well be costly or impractical, thus it is attractive to first rank the collections in the database, in decreasing order of a metric that measures how useful a collection is likely to be for that query. The user can then query the top-ranked collections directly, or the meta-search engine can do so without user intervention. Ranking the collections then provides a form of *collection selection*.

Several techniques have been proposed for collection selection. There are two general kinds of technique based on indexing. One is to gather the set of distinct terms, or *lexicon*, from each collection, together with some inter- and intra-collection statistics such as the number of documents containing each term; these statistics may then be used to estimate the appropriateness of each collection to a query. For large collections, such a lexicon is typically around 1% of the size of the documents it describes, but relative size increases as database size falls. Since the bulk of the space is the distinct words, and since many words are repeated between collections, as a rough guide a selection index is no larger than the space required for the distinct words. The well-known CORI (Callan, Lu, & Croft, 1995) method is based on lexicons.

The other kind of selection technique is to index documents in the database of collections, then select at query time the collection with the most promising documents. Fully indexing each document is expensive—and in effect yields a monolithic implementation—so some compromise is required. One is to select representative documents from each collection, a method that is implicit in clustering and centroids (Salton & McGill, 1983), which to our knowledge has not been evaluated. Another is to index a surrogate representation, or summary, of each document. Standard document summaries are produced for human consumption and are not an ideal surrogate for this task; they include stop words, for example. A surrogate for indexing purposes can be created by methods such as selecting the k highest-weight terms from the document, a method also used for choosing terms for query expansion. We have explored such strategies in previous work, but did not find them superior to using lexicon-based methods.

These methods assume that the systems cooperate, an assumption that is made throughout this paper. Other approaches to distributed retrieval are closer to the model used in Web meta-search systems. In principle, particularly in the absence of cooperation, it would be possible for a collection to contain many relevant documents, be highly ranked, but for the collection's ranking

IPR2024-01306
UIPATH EXHIBIT 1026

mechanism to be unable to find the documents. In practice, in cooperative systems in which information such as term statistics are shared, this is no more likely than in a monolithic system (de Kretser, Moffat, Shimmin, & Zobel, 1998). Thus it is reasonable to compare systems by their ability to find collections with relevant documents, while noting that there remains the issue of how to find documents within collections and combine these results.

A shortcoming of some previous research in this area is that the comparison of selection methods has been on test collections that do not represent likely patterns of distribution in real multi-collection environments. In some of the TREC-based evaluations, for example, the collections were created by division into large sets of similar size (Callan et al., 1995; Voorhees, 1996; Voorhees et al., 1995). More seriously, some comparison has been on collections in which a monolithic collection has been divided in a way that is somewhat artificial, where the allocation of documents to collections has either been random, or based on chronology; with the TREC newspaper articles, for example, one common subdivision has been month by month. These collections are unlikely to show a great deal of topic specificity, other than the fraction of the articles that follow a range of threads of topical interest. For meta-search on the web, where each collection is a web site, such a division is highly improbable, as is evidenced by the TREC-8 Small Web Track data source (Hawking, Voorhees, Crasswell, & Bailey, 2000) used by Crasswell, Bailey, and Hawking (2000) in their evaluation of web-based server selection.

Another way of dividing documents into the test collections is by the way they are used or created. Categorisation techniques can be used to divide documents among a limited number of areas but such an approach assumes that the documents are created centrally, then distributed. Furthermore, to date there has been no application of categorisation to the collection selection problem (Sebastiani, 2002). An alternative is to use attributes of the documents that reflect their origin: for example, their authors, or the place they were written, or some other simple organisational protocol. We believe such an organisation of collections more closely reflects actual workplace practices and provides a valuable contrast to other test collections. We call a group of organised collections a *managed distributed document database* because documents are assigned to collections in some more-or-less systematic fashion. Examples include databases where the collections are individual web sites; databases where each collection is the set of documents created at a particular office (such as a branch of a government department) or by a particular person (such as a journalist); or databases where each collection is drawn from a particular division of an organisation. To our knowledge only one previous paper, by Larkey, Connell, and Callan (2000), explicitly explores managed collections; unfortunately, as discussed later, their results are flawed by a serious methodological error. While some of the earlier work with dividing the TREC data into collections, such as that of Voorhees, Gupta, and Johnson-Laird (1995), can be described as managed, the few collections used and the fact that they were created to be of the same size means that they are unlikely to be representative of real data.

In this paper we re-evaluate a range of collection selection techniques on two sets of collections, where the documents are distributed into around 500 and 2000 collections respectively. The documents are distributed both by methods used previously (randomly and by chronological ordering) and by a document attribute reflecting their origin (such as document author). These experiments produce two major results. One is that distribution by attribute allows more effectively selection of collection, or, alternatively, is better at placing relevant documents together, despite the simplicity of the distribution method and the large number of collections involved. The

IPR2024-01306
UIPATH EXHIBIT 1026

other is that, on a managed collection, selection by document surrogate is clearly superior to selection by lexicon methods.

Our experiments also pose questions about previous results in the area. With a chronological division into collections, the differences between the various methods are small, whether the queries are short or long. Indeed, on these collections many of the methods we tested do little better than the fixed strategy of choosing the largest collection first. When the collections are all of similar size (another unrealistic aspect of many earlier experiments), differences in performance are even less conspicuous. In our experiments, most of the differences in performance observed on these collections are not statistically significant. A further observation concerns long queries, which, while not widely used in some contexts (Spink, Wolfram, Jansen, & Saracevic, 2001), are for example generated by query expansion and moreover provide an interesting point of comparison. Not only are long queries better at collection selection—which in itself is unsurprising—but reveal even greater differences between the methods, and thus provide a way of discriminating between methods.

A particular result is that our experiments show that for managed data CORI is markedly inferior to the other methods tested. To the best of our knowledge previous work has not explored a comparative study of CORI and the lexicon methods presented in this paper. However, since CORI has been widely reported as an effective collection selection method, its poor performance in these experiments is deeply surprising.

2. Collection selection

The collection selection problem has been widely investigated. In most of the techniques that have been proposed, each collection is ranked according to a *goodness* score, computed from a selection index. The score is a measure of how likely collection c is to contain documents that are relevant to query q. Collections are ranked according to their goodness score, and the selected (candidate) collections are subsequently interrogated with the given query, and the matching documents retrieved. The problem of collection selection is, thus, how best to compute a score for a collection.

2.1. Lexicon methods

Most approaches for collection selection use indexes based on the collections' lexicons. That is, they use similarity functions based on the collection vocabularies (with term statistics). The selection index contains the combined lexicon of the collections, together with, for each term t, the number of collections it occurs in and its frequency $f_{c,t}$ in each collection c.

Early examples of lexicon schemes were the *GlOSS* (Gravano, Garcia-Molína, & Tomasic, 1994a, 1994b) and *gGlOSS* (Gravano & Garcia-Molína, 1995) systems. Both of these systems maintain, for each term-collection pair, the document count in the collection and the aggregate term weight across all documents in the collection. The retrieval systems use the term weights to estimate the significance of particular terms in collections and queries. (A problem with distributed retrieval is that weights returned by different collections may be computed differently, so that the scores are incomparable.)

IPR2024-01306
UIPATH EXHIBIT 1026

A similar approach is the *CORI* algorithm used in *INQUERY* system (Broglio, Callan, Croft, & Nachbar, 1994; Callan et al., 1995), which also uses lexicon information to rank and select collections via a probabilistic model. Previous work by French et al. (1999) and French, Powell, Viles, Emmitt, and Prey (1998) found that CORI was superior to gGlOSS, and thus we do not test GlOSS/gGlOSS here. We do however investigate CORI, formulated as follows:

CORI: The ranking score for a given collection is the sum of the belief probabilities of query terms appearing in the collection. We used the following formulation, previously used in other comparative collection selection experiments by French et al. (1999) and Callan (2000), to compute the similarity of query q to collection c over the set of terms they have in common:

$$\mathrm{CORI}(q,c) = \frac{\sum_{t \in q \& c}(0.4 + 0.6 \cdot T_{c,t} \cdot I_{c,t})}{N_q}$$

where

$$T_{c,t} = f_{c,t} \Big/ \left(f_{c,t} + 50 + 150 \cdot \frac{U_c}{\sum_{c \in C} U_c / N} \right)$$

represents a modification to the document frequency of the term within the collection and $I_{c,t} = \log((N + 0.5)/N_t)/\log(N + 1.0)$ represents a modification to the inverse collection frequency of the term within the database. The parameters are described in Table 1. [1]

Yuwono and Lee (1997) investigated a lexicon scheme that used for each indexed term its *cue validity variance*, which measures the usefulness of the term for distinguishing one collection from another. This measure has been found to be inferior to CORI in several investigations (Callan, Powell, French, & Connell, 2000; Crasswell et al., 2000) and fared no better against inner product (described below) in a study by D'Souza, Thom, and Zobel (2000), and we do not test it here.

Zobel (1997) investigated a range of lexicon-based ranking algorithms. Variations included the inner product, which Gravano and Garcia-Molína (1995) had previously identified as being effective; query term skew, which is based on measuring whether the query terms are atypically frequent in a collection; and high-similarity cosine, where the highest possible similarity for a document in the collection is computed on the optimistic assumption that all the query terms found in the collection will be found in a single document. The formulations used are as follows. In these formulations, the query-term weight is $w_{q,t} = w_t \cdot \log(f_{q,t} + 1)$, where w_t represents the importance of term t across all collections and is given by $w_t = \log(N/f_t + 1)$. The other parameters are described in Table 1.

InnProd: The collection ranking score is the sum of the products of query term and term collection weights:

[1] The definition of CORI is not completely consistent between papers, and in particular the quantity U_c is either the number of distinct terms or the number of term occurrences. Based on preliminary experiments, we chose to use the former because it gave slightly superior effectiveness.

IPR2024-01306
UIPATH EXHIBIT 1026

Table 1
Parameters used in similarity formulations

$w_{c,t}$	Weight of term t in collection c
$w_{q,t}$	Weight of term t in query q
w_t	Weight of term t across all collections
$f_{c,t}$	Document frequency of term t in collection c, that is, the number of documents in collection c that contain t
$f_{d,t}$	Number of occurrences of term t in document d
$f_{q,t}$	Number of occurrences of term t in query q
f_t	Total number of documents that contain term t
$F_{c,t}$	Number of occurrences of term t in collection c
N_c	Number of documents in collection c
U_c	Number of distinct terms in collection c
C	Set of collections
N	The total number of collections
N_q	The number of distinct terms in query q
N_t	Number of collections containing term t

$$\mathrm{InnProd}(q,c) = \sum_{t \in q\&c} w_{q,t} \cdot w_{c,t}$$

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

Skew: The function is a form of inner product that highly ranks collections where the query terms are atypically frequent:

$$\mathrm{Skew}(q,c) = \sum_{t \in q\&c} \frac{f_{c,t}}{f_t} \cdot \log(f_{q,t} + 1) \cdot w_t$$

HighSim: The function uses the cosine measure to estimate the highest-available similarity for a typical-length document in the collection:

$$\mathrm{HighSim}(q,c) = \frac{\sum_{t \in q\&c} w_{q,t} \cdot w_{c,t}}{W_c}$$

where $w_{c,t} = w_t \cdot \log(F_{c,t} + 1)$ and $W_c = \sqrt{(\sum_{t \in c} F_{c,t})/N_c}$ is the average number of term occurrences in each document in collection c.

More recently, the use of lightweight probes (Hawking & Thistlewaite, 1999) and query-based sampling (Callan et al., 2000; Crasswell et al., 2000) has been explored. These techniques construct the required lexicon information by sending small (single or multiple-term) queries to all collections in the database with the aim of constructing partial lexicons. Previous work by Callan, Connell, and Du (1999) showed that such partial lexicons are representative of complete lexicons for the collections. In the case of lightweight probes, probe queries are sent to each collection at query time and require the cooperation of servers to obtain up-to-date term statistics. Query-based sampling eliminates the need for cooperation, through interaction with collections via probe queries, before query time; the results of such interactions are then used to establish (partial) term statistics for each collection.

IPR2024-01306
UIPATH EXHIBIT 1026

Whatever the approach, Xu and Callan (1998) showed that poor collection selection ultimately impacted on document retrieval effectiveness from distributed databases, and investigated the use of query expansion and other techniques to aid the selection process.

2.2. Surrogate methods

In a partial indexing approach proposed by D'Souza and Thom (1999), only a subset of terms from each document is indexed. This technique extracts a set of n terms from each document, where n is typically fixed at some predetermined value, though in principle it may be varied from document to document. In these *n-term* indexes, only a fraction of the lexicon of each collection is stored. However, for each term it is necessary to store the identifier of every surrogate it occurs in. Surrogate methods require a high level of cooperation between systems, as per-document information must be shared, but have potential advantages in environments where collections are a mix of topics and avoid the need to correct for factors such as collection size.

Several term extraction algorithms could be used. We evaluate two schemes in this paper, known as the *first-n* and *best-n* algorithms. The first-n algorithm extracts the first n unique terms from each document in the database. To implement best-n, we used

$$\log(1 + f_{d,t}) \cdot \log\left(\frac{\sum_{c \in C} N_c}{f_t}\right)$$

to identify the n highest-weight terms from each document, as is done in query expansion. Note that global statistics are used to determine term significance, to give reliability. The value of n can be chosen so that the space required for the index is roughly the same as in the lexicon methods; we found in earlier work that n between 20 and 40 is reasonable. Our choice here to use $n = 20$ is, therefore, conservative.

For any given term extraction regime there are several ways to generate goodness scores to rank the collections. Our approaches exploit the fact the n-term index retains document references, beyond the single statistic maintained in the lexicon schemes. Consider Table 2, which presents a set of ranked documents generated from querying an n-term index. The similarities are query–document similarity scores and are ordered from highest to lowest. The bottom line shows which collection the document was found in. Collection c_{30} is represented at least three times in the table, suggesting that it may be relevant to the query, and, according to the simplistic ranking protocol "use the highest document rank as the goodness score", the score for collection c_{30} is 0.316. However, the other scores for collection c_{30} could also be used.

Five functions were investigated. These are as follows, where C_c is the set of documents ranked in collection c, $\mathrm{sim}(q, d)$ is the similarity of document d and query q according to the cosine measure, r_d is the rank ordinal of document d, and K is a constant that alters the shape of the

Table 2
Sample set of ranked documents and associated similarity measures and collections, for a given query

Document	d_{21}	d_{200}	d_{63}	d_{74}	d_5	d_{126}	d_{722}	d_8	d_{19}	$\cdots$
Similarity	0.316	0.278	0.265	0.261	0.160	0.157	0.141	0.140	0.139	$\cdots$
Collection	c_{30}	c_{43}	c_{72}	c_{69}	c_{72}	c_{30}	c_{51}	c_{35}	c_{30}	$\cdots$

IPR2024-01306
UIPATH EXHIBIT 1026

distribution, arbitrarily set to 10. Note that where $\mathrm{sim}(q, d)$ computations appear in formulations these are computed globally, and are therefore immune to collection dependencies.

Naive: The highest-ranked collection has the highest ranked document, that is, the document with the highest similarity for query q:

$$\mathrm{Naive}(q, c) = \max_{d \in C_c} \mathrm{sim}(q, d)$$

InvRank: The highest-ranked collection has the largest aggregate inverse document ordinal in the rankings list:

$$\mathrm{InvRank}(q, c) = \sum_{d \in C_c} \frac{1}{r_d + K}$$

SimDivRank: The highest-ranked collection has largest aggregate of query–document similarity divided by document ordinal in rankings list:

$$\mathrm{SimDivRank}(q, c) = \sum_{d \in C_c} \frac{\mathrm{sim}(q, d)}{r_d}$$

SumSim: The highest-ranked collection has largest aggregate similarity, where the aggregate similarity is the sum of similarities for documents appearing in the collection:

$$\mathrm{SumSim}(q, c) = \sum_{d \in C_c} \mathrm{sim}(q, d)$$

SumSimSqr: As for *SumSim* but with aggregate similarity squared:

$$\mathrm{SumSimSqr}(q, c) = \sum_{d \in C_c} \mathrm{sim}(q, d)^2$$

All but the first of these require in principle a total ranking of the documents in all collections, but in practice, for all methods but SumSim, the numbers quickly become small and only the first few hundred top-ranked documents need be considered.

3. Evaluation of collection selection algorithms

In evaluating the performance of collection selection algorithms, the question that needs to be addressed is: How good is an algorithm at ranking collections such that the number of relevant documents returned is maximised? The question of evaluation is a choice between selecting collections that contain highly similar documents or those that contain highly relevant documents (Zobel, 1997). If the aim is to emulate a retrieval mechanism, then the former choice is appropriate; this was the evaluation methodology used for the GlOSS/gGlOSS systems (Gravano & Garcia-Molína, 1995; Gravano et al., 1994a, 1994b). However, this approach does not minimise the number of collections to be interrogated, as a small collection with a couple of highly similar documents will be ranked ahead of a large collection with many documents of slightly lower similarity.

IPR2024-01306
UIPATH EXHIBIT 1026

In a comparative study of high similarity versus high relevance evaluation methodologies, French et al. (1999) showed that the gGlOSS systems are poor compared to other systems, such as CORI, when measured with high-relevance evaluation, confirming that the GlOSS/gGlOSS evaluations were inaccurate. Furthermore, French et al. (1999) found CORI to be superior to gGlOSS for a range of evaluation metrics. In this paper we evaluate performance using a relevance-based evaluation methodology.

However, a further major factor that has not been widely investigated is the mechanism used to allocate documents to collections. It could be argued that the allocation method may have only limited impact on results: while, for example, one method of distributing documents may allow better effectiveness, the relative performance of the methods may not change. In our view, however, this assumption is unwarranted.

We argue that methods for distributed retrieval should be evaluated on test collections that reflect the way in which distributed retrieval might actually be used. It is unlikely to be the case that individual document repositories consist of distinct, randomly chosen documents, for example. In practical distributed retrieval, each repository is likely to have a theme, topic, or owner. Similar conditions should be simulated in evaluations. We therefore propose measurement over *managed* collections, where the documents in each individual repository have some property in common—in particular, properties that give the collections some cohesiveness of topic.

Larkey et al. (2000) have compared performance over managed collections to performance on more artificial collections. However, their work has a serious flaw. The chronological collections are of almost identical size; the sizes in one set range from 3461 to 3486 documents, while in other they range from 7294 to 7637 documents. In the managed collections, the sizes range from 1 to 34,271 documents and from 100 to 100,782 documents respectively. In such circumstances, simply choosing the largest collection first can give much greater effectiveness, and repartitioning the chronological documents on the same distribution would also greatly improve effectiveness. The managed and chronological collections are incomparable, and conclusions as to the advantages of a topic-specific partition cannot be drawn from their experiments.

Nonetheless, it was our expectation that use of managed collections had the potential to change the relative performance of the measures we were investigating. With topic-specific collections, the number of terms per collection would fall, particularly with surrogate or n-term methods, and the different methods would have different sensitivity to the noise introduced by out-of-topic documents.

Previous studies evaluating collection selection (and document retrieval) have used collections where the documents are distributed either randomly or using some form of chronological ordering. Important work prior to 1998 is summarised by French et al. (1998). More recent efforts have typically used chronologically ordered collections (French et al., 1998, 1999; Powell et al., 2000) where the collections were roughly equi-sized either in terms of storage or number of documents. The main exceptions have been the work of Larkey et al. (2000) as discussed above, and the GlOSS/gGlOSS experiments, where Usenet newsgroups were used, sources that are a clear instance of a managed database. However, there are no relevance judgements for the newsgroups, and thus relevance-based assessment cannot be made. Furthermore, a relatively small number of collections was used, as indeed has been the case in the afore-mentioned studies that used relevance-based rankings. Exceptions to these are studies such as reported by Callan et al. (2000) and Crasswell et al. (2000). In the former case, however, the 921 collections were roughly of equal size;

IPR2024-01306
UIPATH EXHIBIT 1026

in the latter study, a more representative distribution of documents was employed, with over 956 web-based data sources. However, none of these studies compared the selection algorithms within the context of using different distribution protocols over the same data sources.

The primary aim of this work was to investigate whether relative performance of the methods depended on the method of distributing documents amongst collections. We considered three distribution policies: random, chronological, and managed. Secondary aims were to evaluate the impact on effectiveness of the different policies, and to determine whether the methods worked when large numbers of collections were involved.

4. Experiments

To test our hypothesis that collection selection is influenced by the way documents are organised within a distributed document database, we ran experiments on six distinct databases, each derived from the same original set of documents. All documents were sourced from the Associated Press TREC volumes 1 and 2 (Harman, 1995). All these documents are in SGML and contain at least a DOCNO field, and optionally one or more BYLINE fields and DATELINE fields. A (heavily edited) portion of such a sample document is as shown in Fig. 1.

To create the first group of three test collections, the set of Associated Press documents was partitioned in three different ways. We chose the BYLINE field to distribute documents into collections to give a managed database. The BYLINE field is an authorship entry that identifies individuals (and agencies) responsible for creating the document; the example document was written by Hillel Italie, an Associated Press Writer, for instance. In the managed database context we could view this document as belonging to the collection of Italie articles. However, BYLINE strings are not always so straightforward, and a filter was used to transform BYLINE strings so that they could reliably be used to identify the document's resident collection. The filter discarded punctuation and noise words such as "by"; standardised case; and selected the first author's name where a list of authors appeared, or the last three words otherwise. (We have not investigated the case where documents are allocated to multiple collections, thus only the first BYLINE was used.)

```
<DOC>
<DOCNO> AP890101-0001 </DOCNO>
<FILEID>AP-NR-01-01-89 2358EST</FILEID>
<HEAD>You Don't Need a Weatherman ... </HEAD>
<HEAD>Eds: Also in Monday AMs report.</HEAD>
<BYLINE>By HILLEL ITALIE</BYLINE>
<BYLINE>Associated Press Writer</BYLINE>
<DATELINE>Grant Editor, NEW YORK (AP) </DATELINE>
<TEXT>
   The celluloid torch has been passed to ...
</TEXT>
</DOC>
```

Fig. 1. Partial document from the TREC data.

IPR2024-01306
UIPATH EXHIBIT 1026

The resulting string was used as a collection name. Documents with no BYLINE fields were rejected and were not included in any collection.

This process yielded 2239 collections, where collection c_i had n_i documents. An initial document population of 164,597 was reduced to 88,062. There was one relatively large collection of 6440 documents, 156 collections ranging in size from 100 to 698 documents, and 2008 collections with size range 1–99, with the number of collections with 1, 2 and 3 documents being, respectively, 731, 174 and 103.

To generate a chronological database, we used the date information in the DOCNO field to order the documents, and divided them into 2239 collections such that collection c_i had n_i documents, where n_i was as for the BYLINE distribution. The distribution profile of collection sizes was therefore the same as for the managed collection. Again, the documents with no BYLINE fields were omitted. To generate a random database, we again used the same profile, but after placing the documents in a random order.

To get our second group of three databases, we repeated the process for the DATELINE field, which represents a dispatch location for the document in question; the example document belongs to the New York collection, for instance. That is, in the managed database context we might view this document as emanating from the AP New York collection. Again, DATELINE strings were not always straightforward, and we used a simple filter to extract a simplified string that could be used to identify a collection. The filter discarded any punctuation and then chose the phrase following the last comma; case was standardised in the resulting words to give a single string. Documents with no DATELINE fields were rejected and did not participate in any collection. This gave us 530 collections with a total population of 153,020 documents, again with a high skew in the individual populations; there were two large collections of 16,248 and 30,507 documents respectively, and 368 collections with less than 100 documents each. As before, we built chronological and random databases with the same collection profile.

We primarily discuss the BYLINE results, but present a summary table for both sets of test databases.

TREC topics 51–200 were used to generate short and full queries for the experiments, with the number of terms per short and full queries averaging, respectively, approximately 4 and 78. Relevance judgements were used to discard queries with no relevant documents in the database, yielding 148 queries from the original set of 150. The total number of relevant documents in the BYLINE collection over these queries was 11,085.

Evaluation of the two indexing schemes for collection ranking involved determining the average number (over all queries) of relevant documents retrieved in the top k ranked collections, where k ranged from 1 to the number of collections (that is, 2239 in the BYLINE case and 530 for DATELINE). Two baselines for evaluation, denoted *fixed* and *perfect*, were established. The fixed baseline ranked the collections according to size, the rationale being that the largest collection was most likely to have more relevant documents than any other—any (useful) indexing technique ought to perform better than the fixed baseline. The perfect baseline was established by reordering the collections according to known relevance counts and averaged over all queries. This evaluation assumes, as does most recent work in this area, that the best collection to select first (rank highest) is the collection containing the most documents that are relevant. Whether this corresponds to maximising the number of relevant documents returned is an open question, especially if the collection sizes vary widely.

IPR2024-01306
UIPATH EXHIBIT 1026

4.1. Results

We ran comprehensive experiments, testing every method on every data set while measuring recall at k collections returned for all valid k. Detailed results are presented below. Some general trends were as follows. For n-term indexing, best-n consistently outperformed first-n. For both query types, InvRank and SumSimSqr were the best n-term methods, with SimDivRank a close third for short queries. In the graphs discussed below, we limit our presentation of n-term methods to InvRank and SumSimSqr. Among the lexicon methods, HighSim, CORI, and InnProd were the best, so these were chosen for presentation in the graphs.

4.1.1. Chronological collections

An environment that corresponds to that used in many previous experiments is the chronological databases with short and long queries. Results for the main methods discussed earlier are shown in Fig. 2. (Results for the other methods are discussed below.) The upper of these figures, for short queries, does not show a clear trend. The lexicon method CORI appears to be the poorest, but another lexicon approach, HighSim, is the best. None of them is dramatically better than the simplistic baseline of largest collection first—a result that is perhaps unsurprising given the size of the largest collection! In the lower graph, however, a rather different picture emerges. CORI is clearly the worst approach, doing no better than the baseline. The n-term (or NTI) methods have been able to get much greater gains from the additional query terms, with best-n outperforming first-n.

We used the Wilcoxon signed-ranked test to establish whether the differences between the collection selection methods were significant. Considering the results for recall at $k = 10$ at the 99% confidence level, no difference was observed between HighSim and SumSimSqr, and no difference was observed between CORI, InvRank, and InnProd. However, both the methods in the first group were superior to all the methods in the second group. For long queries, InvRank and SumSimSqr are indistinguishable, but all other differences were significant.

Note that all methods are about equally good for recall at $k = 1$, that is, at the first collection retrieved. In almost all cases, the largest collection was the highest match, regardless of measure used.

Given that previous work with CORI has consistently shown that it works well on chronological data, it is surprising that it has performed badly here. One possibility is that it is poor at selecting very small collections, where the number of distinct terms is high compared to the number of documents. Another is that it may be poor because the single large collection effectively diminishes the U_c term in the formulation. Another is that it does not appear to have previously been compared to these measures.

We strongly suspect that the parameter settings (which yield the constants of 50 and 150 in the formulation) are inappropriate for databases with a large range of collection size. These constants were determined by tuning on a database where the number of collections was small and they were of similar size to within an order of magnitude.

4.1.2. Random collections

Similar trends to those discussed above can be observed in Fig. 3. Again, for short queries the methods are barely distinct, while with long queries the n-term (that is, surrogate) methods

IPR2024-01306
UIPATH EXHIBIT 1026

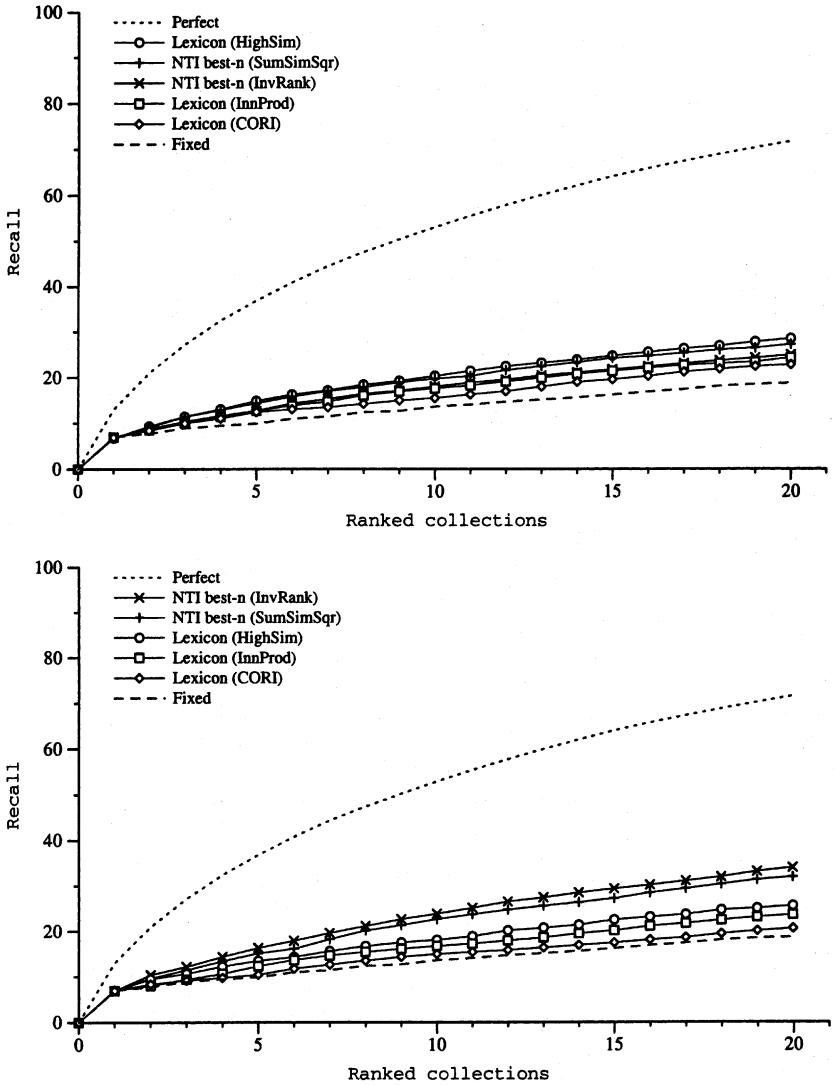

Fig. 2. Chronological BYLINE. Cumulative recall of NTI best-n (InvRank, SumSimSqr) versus lexicon (HighSim, InnProd, CORI) for top k collections ($k \leqslant 20$). Top graph: short queries. Lower graph: full queries.

outperform the lexicon methods. Even the best method only produces limited improvements over the largest-collection-first baseline.

Significance tests identified only a few differences. For long and short queries, in a few instances CORI was inferior to the other methods, and HighSim was consistently superior to the other lexicon methods. However, only HighSim was consistently superior to the elementary fixed method of taking the largest collection first—for most of the other methods, no significant difference was observed. This is strong evidence that collections with no topic specificity are useless for evaluation of collection selection techniques.

IPR2024-01306
UIPATH EXHIBIT 1026

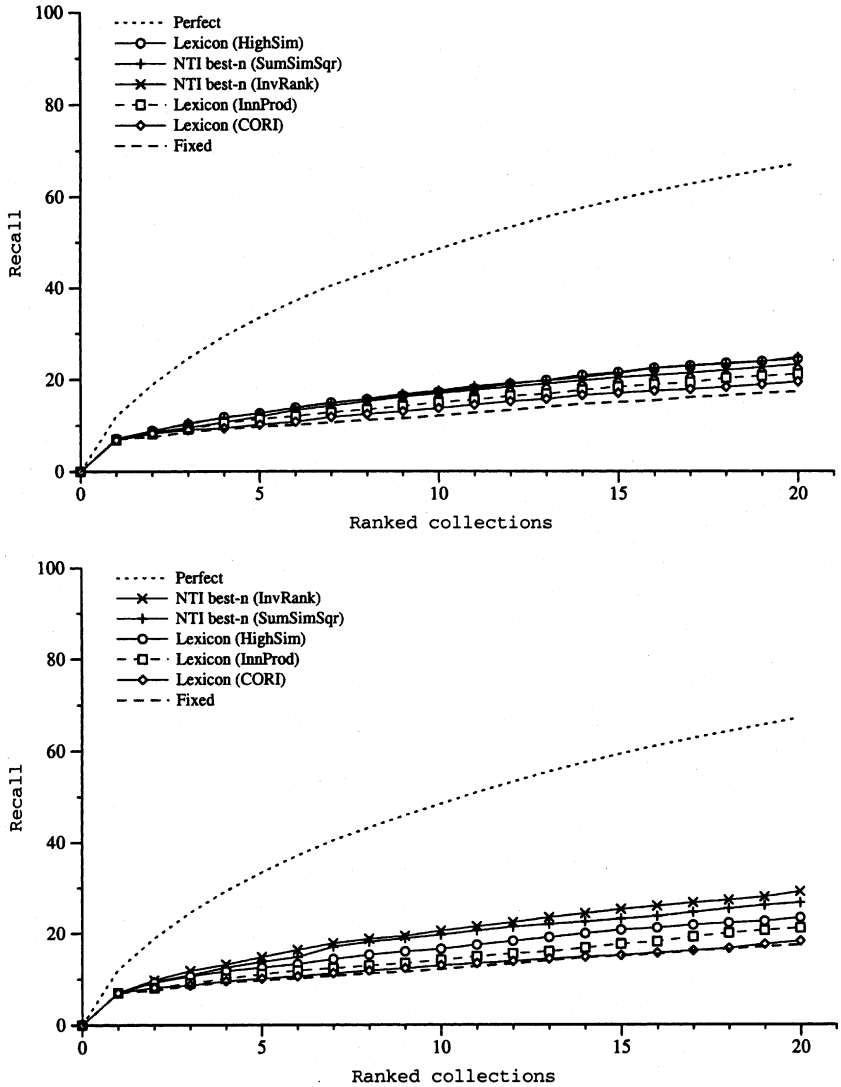

Fig. 3. Random BYLINE. Cumulative recall of NTI best-n (InvRank, SumSimSqr) versus lexicon (HighSim, InnProd, CORI) for top k collections ($k \leqslant 20$). Top graph: short queries. Lower graph: full queries.

4.1.3. Managed collections

Results for managed collections are shown in Fig. 4. Several changes are immediately obvious. One is that "largest collection first" is, relatively, much less effective, and that it is possible to achieve considerably better recall at $k = 1$ by choosing other collections. Presumably this is because some of the collections are fairly topic specific (the same journalist or office writing on the same topic) and these topics match queries well. Another change is that all the methods manage to do much better than the "largest first" baseline.

A further change is that there is more difference between the retrieval methods. For short queries and recall at $k = 5$, the best method, InvRank, scores around 31, compared to around 23

IPR2024-01306
UIPATH EXHIBIT 1026

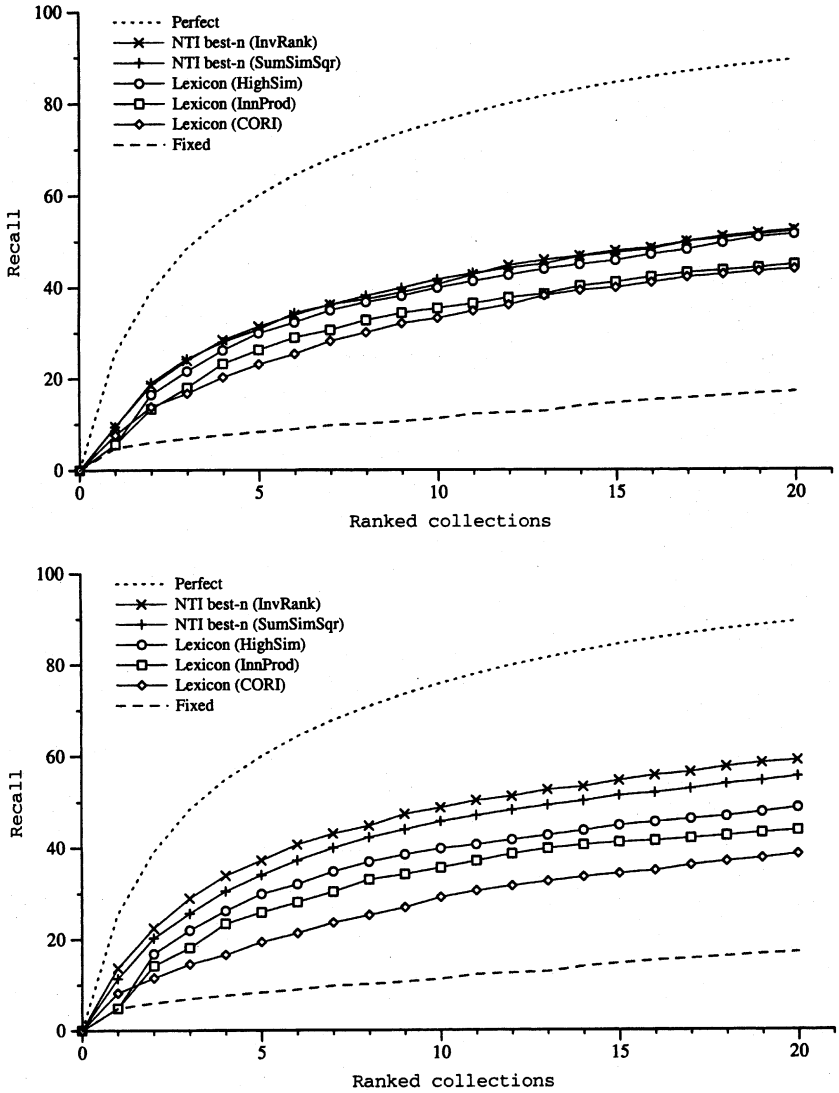

Fig. 4. Managed BYLINE. Cumulative recall of NTI best-n (InvRank, SumSimSqr) versus lexicon (HighSim, InnProd, CORI) for top k collections ($k \leqslant 20$). Top graph: short queries. Lower graph: full queries.

for CORI. For long queries, CORI's performance falls to around 18, while that of InvRank rises to around 38. These are dramatic differences.

We again used the Wilcoxon measure to investigate significance. On short queries, no significant difference was observed between HighSim, InvRank, and SumSimSqr, nor was a difference observed between CORI and InnProd. However, the first group was superior to the second group. For long queries, all differences were significant; this data set was the only one in which the numerical results were fully confirmed statistically.

The problem with CORI may simply be the lack of use of query-term weights, or may reflect some underlying issue such as the need for tuning to a particular collection. However, as it stands,

IPR2024-01306
UIPATH EXHIBIT 1026

it is unlikely that, for example, it would be beneficial to use some form query expansion in conjunction with CORI. In contrast, for the other methods it is apparent that additional terms can greatly improve effectiveness.

4.1.4. BYLINE and DATELINE summaries

The complete set of experiments with the BYLINE databases is summarised in Table 3, which presents the recall at $k = 10$ for these experiments.

Table 4 presents corresponding results for the DATELINE collections, over the same set of experiments. The results were broadly consistent with BYLINE, but the distinction between the *n*-term and lexicon schemes was reduced. Once again, for *n*-term, best-*n* was more effective than first-*n*, and the best *n*-term indexing was more effective than lexicon methods.

In Table 4, the margins were small. For short queries on the managed collections, CORI, the worst method, was only 5.6% worse than SumSimSqr, the best. For the chronological and random versions of the data, all methods were barely distinguishable. These results do however confirm our main contention: that experiments on a chronological division into collections are unrevealing. On the managed collections, it is clear that the *n*-term methods are superior to the lexicon methods, and it seems that the way in which the terms are selected does not have great impact.

Table 3
Recall @ 10 for BYLINE experiments

	Short queries			Full queries		
	Managed	Chrono	Random	Managed	Chrono	Random
Baselines:						
Perfect	75.83	52.89	48.44	75.83	52.89	48.44
Fixed	11.22	13.60	12.12	11.22	13.60	12.12
Lexicon:						
HighSim	39.83	20.29	17.29	39.79	18.09	16.52
InnProd	35.38	17.58	15.02	35.60	16.74	14.09
Skew	34.37	17.72	14.74	33.10	15.89	14.59
CORI	33.20	15.47	13.76	29.13	14.96	12.96
NTI (best-n):						
Invrank	40.56	17.98	16.97	48.73	23.85	20.52
Naive	29.05	11.96	12.12	42.71	17.64	17.08
Simdivrank	39.83	17.40	16.51	47.38	21.70	19.58
SumSim	38.48	18.16	16.25	39.80	18.05	14.66
SumSimSqr	41.67	19.69	17.57	45.72	22.67	19.61
NTI (first-n):						
Invrank	38.24	19.27	16.90	46.89	21.00	19.55
Naive	29.30	13.30	12.58	40.94	16.62	15.66
SimDivRank	37.48	18.13	16.49	45.36	19.31	18.17
SumSim	35.61	15.48	14.76	33.62	15.22	13.67
SumSimSqr	37.94	18.05	17.36	41.85	18.66	16.02

IPR2024-01306
UIPATH EXHIBIT 1026

Table 4
Recall @ 10 for DATELINE experiments

	Short queries			Full queries		
	Managed	Chrono	Random	Managed	Chrono	Random
Baselines:						
Perfect	89.72	69.77	65.79	89.72	69.77	64.89
Fixed	59.80	48.52	48.72	59.80	48.52	48.02
Lexicon:						
HighSim	69.76	49.67	48.94	67.33	49.82	48.50
InnProd	68.05	49.36	48.85	66.40	49.12	48.17
Skew	67.56	49.11	47.23	66.12	47.43	47.48
CORI	65.65	49.16	47.90	64.64	48.65	48.02
NTI (best-n):						
Invrank	70.18	49.32	47.46	73.74	51.61	49.75
Naive	56.81	29.39	28.71	69.51	41.39	39.58
SimDivRank	69.53	47.26	45.63	72.82	48.65	45.35
SumSim	69.38	50.13	48.98	69.52	49.39	48.20
SumSimSqr	71.24	50.43	49.22	72.10	50.99	48.88
NTI (first-n):						
Invrank	69.09	47.93	46.31	72.06	50.11	49.39
Naive	56.39	30.69	30.42	68.13	39.90	38.94
SimDivRank	67.15	45.69	44.41	72.00	47.10	45.62
SumSim	67.19	50.04	48.57	66.14	48.57	48.11
SumSimSqr	68.18	50.61	49.04	69.08	49.36	48.20

4.2. Further experiments

Some of the earlier work in this area was done on a set of 43 collections developed in the context of TREC (Voorhees, 1996). (The data on disks 2 and 4 was divided into 98 collections; of these 43 collections came from disk 2.) These collections are of approximately similar size, and are not specific in topic.

We used this data set to examine the behaviour of the different methods, using both short and full queries. As we had found in our earlier work, and as observed above, n-term methods are no better than the lexicon methods on this data. However, few of the differences are statistically significant, and therefore this data set—or any, we believe, of this design—has little value as a testbed for evaluating collection selection techniques.

5. Conclusions

We have compared a range of index-based collection selection techniques on several databases. These experiments contrast the well-known CORI lexicon-based method with other lexicon-based and surrogate-based methods. They also contrast both the performance available and the changes in behaviour observed for different approaches to constructing test collections.

IPR2024-01306
UIPATH EXHIBIT 1026

Our results are unequivocal. Of the methods tested, CORI is clearly the weakest, coming last or near-last in all 12 comparisons. The best surrogate methods were overall markedly superior to the lexicon methods, particularly on managed collections, and with greater improvements for long queries than for short. Thus query expansion may further improve the surrogate methods. In contrast, the lexicon methods did not gain effectiveness when additional query terms were available.

It is also clear that managed collections provide a very different test environment to that given by collections with a chronological or random division. On the latter, the differences between the methods was small, and they generally did little better than a trivial benchmark. On the former, much greater effectiveness could be gained, and much greater differences were observed between the methods. It is our view that managed collections reflect one likely way in which distributed retrieval would be used in practice, with for example authors maintaining repositories of their own work.

Our results show that surrogate methods are the best way of selecting collections in such databases. Despite the strong performance reported for CORI in other work, we consistently found that it was not effective. Our results also show that the kinds of collections widely used in previous work for evaluating collection selection techniques are not a reliable testbed for a likely class—arguably, the most likely class—of distributed retrieval problems. They do not appear to allow significant differences to be observed between methods.

An alternative approach is to investigate databases where the documents are distributed by an automatic categoriser. Although this represents another way in which distributed retrieval might be used, and the number of categories used in current work is relatively small, our results show that topic-based distribution does allow more effective selection; thus surrogate methods should be considered for distributed retrieval on categorised databases.

Acknowledgements

This work used computing facilities supported by the Australian Research Council. We would also like to thank the anonymous referees, whose comments we found helpful in revising this paper.

References

Broglio, J., Callan, J., Croft, W. B., & Nachbar, D. (1994). Document retrieval and routing using the INQUERY system. In D. Harman (Ed.), *Proceedings of third text retrieval conference (TREC-3)* (pp. 29–38). National Institute of Standards and Technology Special Publication 500-225, Washington.

Callan, J. P. (2000). Distributed information retrieval. In W. B. Croft (Ed.), *Advances in information retrieval: recent research from the center for intelligent information retrieval* (pp. 127–150). Kluwer Academic Publishers.

Callan, J., Connell, M., & Du, A. (1999). Automatic discovery of language models for text databases. In *Proceedings of the 1999 ACM international conference on management of data (SIGMOD'99)* (pp. 479–490). ACM.

Callan, J. P., Lu, Z., & Croft, W. B. (1995). Searching distributed collections with inference networks. In E. A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 21–28). ACM.

Callan, J., Powell, A. L., French, J. C., & Connell, M. (2000). *The effects of query based sampling in automatic database selection algorithms*. Technical Report CMU-LTI-00-162, Language Technologies Institute, School of Computer Science, Carnegie Mellon University.

IPR2024-01306
UIPATH EXHIBIT 1026

Crasswell, N., Bailey, P., & Hawking, D. (2000). Server selection on the world wide web. In *Proceedings of the fifth ACM conference on digital libraries* (pp. 37–46).

de Kretser, O., Moffat, A., Shimmin, T., & Zobel, J. (1998). Methodologies for distributed information retrieval. In M. P. Papazoglou, M. Takizawa, B. Kramer, & S. Chanson (Eds.), *Proceedings of the eighteenth international conference on distributed computing systems* (pp. 66–73). Amsterdam, the Netherlands.

D'Souza, D., & Thom, J. A. (1999). Collection selection using *n*-term indexing. In Y. Zhang, M. Rusinkiewicz, & Y. Kambayashi (Eds.), *Proceedings of the second international symposium on cooperative database systems for advanced applications (CODAS'99)* (pp. 52–63). Wollongong, NSW, Australia: Springer.

D'Souza, D., Thom, J. A., & Zobel, J. (2000). A comparison of techniques for selecting text collections. In *Proceedings of the eleventh Australasian database conference (ADC'2000)* (pp. 28–32). Canberra, ACT, Australia. To appear in IEEE Computer Society publication.

French, J. C., Powell, A. L., Callan, J., Viles, C. L., Emmitt, T., Prey, K. J., & Mou, Y. (1999). Comparing the performance of database selection algorithms. In M. Hearst, F. Gey, & R. Tong (Eds.), *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval* (pp. 238–245). ACM.

French, J. C., Powell, A. L., Viles, C. L., Emmitt, T., & Prey, K. J. (1998). Evaluating database selection techniques: a testbed and experiment. In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, & J. Zobel (Eds.), *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 121–129). ACM.

Gravano, L., & Garcia-Molína, H. (1995). Generalising GlOSS to vector-space databases and broker hierarchies. In *Proceedings of 21st international conference on very large data bases* (pp. 78–89). September 11–15, Zurich, Switzerland.

Gravano, L., Garcia-Molína, H., & Tomasic, A. (1994a). The effectiveness of GlOSS for the text database discovery problem. In *Proceedings of SIGMOD 94* (pp. 126–137). ACM.

Gravano, L., Garcia-Molína, H., & Tomasic, A. (1994b). Precision and recall of GlOSS estimators for database discovery. In *Proceedings of 3rd international conference on parallel and distributed information systems* (pp. 103–106). Austin, TX: IEEE-CS.

Harman, D. (1995). Overview of the second text retrieval conference (TREC-2). *Information Processing and Management, 31*(3), 271–289.

Hawking, D., & Thistlewaite, P. (1999). Methods for information server selection. *ACM Transactions on Information Systems, 17*(1), 41–76.

Hawking, D., Voorhees, E., Crasswell, N., & Bailey, P. (2000). Overview of the TREC-8 web track. In E. Voorhees, & D. Harman (Eds.), *Proceedings of eighth text retrieval conference (TREC-8)* (pp. 131–148). National Institute of Standards and Technology Special Publication 500-246, Washington.

Larkey, L., Connell, M., & Callan, J. (2000). Collection selection and results merging with topically organized U.S. patents and TREC data. In *Proceedings of the ninth ACM international conference on information and knowledge management (CIKM'00)* (pp. 282–289).

Meng, W., Yu, C., & Liu, K.-L. (2002). Building efficient and effective metasearch engines. *ACM Computing Surveys, 34*(1), 48–89.

Powell, A. L., French, J. C., Callan, J., Connell, M., & Viles, C. L. (2000). The impact of database selection on distributed searching. In *Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval* (pp. 232–239). ACM.

Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval*. New York: McGraw-Hill.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*(1), 1–47.

Spink, A., Wolfram, D., Jansen, M. B., & Saracevic, T. (2001). Searching the web: the public and their queries. *Journal of the American Society of Information Science and Technology, 52*(3), 226–234.

Voorhees, E. (1996). The TREC-5 database merging track. In E. Voorhees, & D. Harman (Eds.), *Proceedings of fifth TREC text retrieval conference* (pp. 103–104). National Institute of Standards and Technology Special Publication 500-238, Washington.

Voorhees, E. M., Gupta, N. K., & Johnson-Laird, B. (1995). Learning collection fusion strategies. In E. A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 172–179).

IPR2024-01306
UIPATH EXHIBIT 1026

Xu, J., & Callan, J. P. (1998). Effective retrieval with distributed collections. In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, & J. Zobel (Eds.), *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 112–120). ACM.

Yuwono, B., & Lee, D. L. (1997). 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)* (pp. 41–49). Melbourne, Vic., Australia.

Zobel, J. (1997). Collection selection via lexicon inspection. In *Proceedings of the 2nd Australian document computing symposium (ADCS'97)* (pp. 74–80). Melbourne, Vic., Australia.

IPR2024-01306
UIPATH EXHIBIT 1026