Supporting User Selection of Digital Libraries

Helen Dodd

Department of Computer Science, Swansea University, United Kingdom

Abstract. This research aims to support users in identifying collections (e.g. digital libraries) that are authorities on the topic they are searching for. These collections should contain a large proportion and quantity of relevant documents, such that they may serve both current and (related) future information needs. This paper presents our research goals for this search task, and the steps taken thus far to achieve them. In addition, we provide our plans for future research in this area.

1 Introduction

Consider a scenario where a user wants to identify authoritative collections (e.g. digital libraries) on a particular topic, or communities publishing related work. They will frequently return to these collections with queries related to the original topic. As such, their goal is to find collections containing a significant *proportion* of relevant material, to fulfil both current and future information needs.

Finding relevant collections is a difficulty faced by some web users [1]. For example, general purpose search engines (e.g. Google) do not support this task. Such engines are focused on document retrieval, where results consist of relevant documents from a large range of sources. To determine collections or websites that are relevant as a whole, the user would need to examine the URLs of the results to identify frequently occurring sources. Therefore, the focus of this research is to develop a search tool to support the user in the above search task: to identify authoritative *collections* that are *about* the user's query.

A technique for identifying collections with relevance to a query is *collection selection*: a tool used to make metasearch more efficient by identifying a subset of collections containing some quantity of relevant documents. The query is then dispatched to these collections, and the results are merged to form a list of relevant documents [10]. In this paradigm, collection selection supports document retrieval, rather than the scenario described above. As such, we will treat collection selection as an independent search task, where the goal is to identify collections that contain a high *proportion* and *quantity* of relevant documents.

The key contributions of this research are to identify whether existing collection selection algorithms and their evaluation techniques can be applied to our collection selection task. Based on our findings we aim to develop an evaluation methodology to suit our needs, and create a new algorithm specifically for our search task. Thus, our hypotheses are as follows:

1. Existing collection selection algorithms will prove to be sub-optimal for our particular selection task.

2. An alternative algorithm can be created, with performance superior to that of the existing algorithms.

This paper presents the work conducted thus far (also documented in [3]) to achieve our research goals. In Section 2 we provide an overview of existing collection selection algorithms and their evaluation metrics. Section 3 describes our proposed approach to evaluating algorithms for our search task; the tools and apparatus we use; and the definition of our new algorithm. We discuss the results of our initial experiments in Section 4, and in Section 5 we set out our plans for our future work.

2 Related Work

This section provides a brief overview of existing collection selection algorithms, and the metrics commonly used to evaluate their performance.

2.1 Algorithms

One of the earliest collection selection algorithms, bGlOSS, predicts the number of relevant documents in each collection, using only the document frequencies of query terms [7]. Therefore, it has low space and computational requirements. Initial results were encouraging, but the tests used only two and six collections, with a significant drop in performance: from 99.04% to 88.95% correct rankings [7] as the collection count increased, which raises concerns of scalability.

Similarly, Cue Validity Variance (CVV) also uses only document frequency data [14], but in contrast calculates the distribution of query terms amongst the collections. CVV was found to outperform CORI (see below), but other studies show the opposite [11].

The CORI algorithm takes a different approach, by treating each collection as a large document (and thus has the same computational complexity as document ranking). As such, it can be summarised as $df.icf$, where df is the number of *documents* containing a query term, and icf is the number of *collections* containing a query term [2]. CORI is often used as a benchmark for evaluating new collection selection algorithms [13, 14], and has been shown to have the most accurate and consistent performance [5, 11]. However, D'Souza et al suggest that CORI is not a viable benchmark due to its use of parameters. Optimal values for these vary over collection sets and even queries; with no obvious method for selecting them prior to execution [4].

Four more algorithms are presented in [16], which make use of *lexicon inspection*. The most effective of these was found to be the Inner Product, which is similar to the vGlOSS similarity function described in [8].

Also of interest is the Decision-Theoretic Framework (DTF) [6], which aims to minimise the cost (e.g. time, money, retrieval quality) of retrieving documents from multiple collections. Whilst the issue of reducing cost is an important one, our work does not deal with it at this time. Our research is currently concerned

with effectively identifying the most relevant collections, rather than considering their associated costs. As such, we do not include DTF in our initial evaluation.

We also consider the suitability of an alternative class of algorithm for our search task; namely *query performance predictors*. These predictors evaluate the quality of results to determine how well the query performs at a search engine. Therefore, when used for collection selection, rankings are based on the predicted quality of results. Several predictors are suggested in [9] and [15]. One of the most effective of these is the Average Inverse Collection Term Frequency (AvICTF) [9], which uses simple term and collection statistics. Thus, we use AvICTF as a representative for query performance predictors in our initial experiments.

2.2 Evaluation Techniques

Collection selection algorithms are commonly evaluated against an *optimal* ranking, such as the Relevance-Based Ranking (RBR) described in [5]. Here, collections are ranked in decreasing order of the number of relevant documents they contain. This requires judgements about which documents are relevant to a query. Often, the algorithms are tested over collections built from TREC data, and so these relevance judgements are readily available. Methods used to determine how well an algorithm ranking estimates the optimal include: Mean-squared error; Spearman rank correlation coefficient; and a recall-based metric [5]. In addition to the optimal, a *baseline* ranking is also used for evaluation. For example, the Size-Based Ranking (SBR) ranks collections in decreasing order of the number of documents they contain [5].

3 Our Approach

In this section we present our approach for evaluating the suitability of algorithms for the task of identifying collections that are relevant to a user's query. In addition, we describe the tools and apparatus we use to support the search task and evaluation process. Finally, we will specify the first version of our own collection selection algorithm.

3.1 Evaluation Technique

To evaluate the suitability and performance of algorithms for our retrieval task we have utilised two techniques: *scenario* based testing, and an *optimal performance* test. We describe these techniques in the following sections.

Scenarios We have developed a scenario based testing strategy to allow us to scrutinise the performance of algorithms over clear cases, with controlled data. They act as a "health check" for the algorithms; those producing incorrect rankings at this stage are unlikely to be suitable for our search task.

We have so far created seven scenarios (described below), in which we vary different attributes of the collections e.g. size, quantity/proportion of relevant

documents, and term frequencies. Each scenario models three collections: C_A, C_B and C_C. Of these, C_A is intended to be the clear winner, and C_C the clear loser:

- S_1: Collections are equal in size. We vary the quantity of relevant documents.
- S_2: Collections differ in size, but the *proportion* of relevant documents (in C_A and C_B) is the same.
- S_3: Collections differ in size, but the *quantity* of relevant documents (in C_A and C_B) is the same.
- S_4: Collections are equal in size. We vary the quantity of relevant documents for a single term query.
- S_5: Collections are equal in size. C_B has the same term occurrences as C_A for some query terms, but does not match all query terms.
- S_6: Collections are equal in size. C_B has higher occurrences of some query terms than C_A, but does not match all query terms.
- S_7: Collections differ in size. C_B is larger than C_A, with a higher quantity (but smaller proportion) of relevant documents. This represents polysemy.

Optimal Performance Test The optimal performance test follows the evaluation strategy described in Section 2.2; where algorithm rankings are examined to determine how well they estimate an optimal ranking. However, rather than the traditional Relevance-Based Ranking, we have developed an alternative optimal that better represents our search goal. For our collection selection task, a top ranked collection should contain a high number of relevant documents, and these should constitute a significant *proportion* of the collection. Therefore, our optimal uses two metrics:

$$RS_c = \frac{|\text{relevant documents in collection}|}{|\text{relevant documents}|} \qquad RP_c = \frac{|\text{relevant documents in collection}|}{|\text{documents in collection}|}$$

where RS_c is the *share* of relevant documents and RP_c is the *proportion* of relevant documents in a collection c. For each query and collection we calculate the harmonic mean (F-score) of these metrics. Thus, our optimal F-score based ranking (FsBR) orders the collections by decreasing harmonic mean. For our initial experiments we use only the Spearman rank correlation coefficient to examine how well the algorithm rankings estimate the optimal.

3.2 Apparatus

A significant proportion of our work thus far has involved the construction of a set of tools to support the performance evaluation of collection selection algorithms. In this section, we describe these tools and other apparatus we use in our experiments.

Doddle The evaluation of collection selection algorithms using the scenario and optimal performance tests is supported by *Doddle*, which consists of two applications. An administration tool enables the management of collection data, and creation of test scenarios. A web service includes interfaces for executing the scenario and optimal performance tests. From these the user can select which algorithms to run; choose an index table; view the results and optimal rankings; and view the Spearman rank correlations between the algorithm and optimal rankings.

OAI-PMH The *optimal performance tests* (see Section 3.1) use real collection data, harvested using the Open Archives Initiative's Protocol for Metadata Harvesting[1] (OAI-PMH). The protocol facilitates the automatic harvesting of document metadata in a collection. For simplicity, we harvest metadata in the mandatory Dublin Core format. We build two indexes: "title only" and "title and description", as title and description fields should be present in the majority of records. We do not harvest full documents due to time and storage costs, and furthermore such access may not be available. One concern of our research is to determine the optimal and minimal metadata to use.

Our use of real collections differs from previous evaluations where test collections are built from TREC data, divided by source and date, or such that collections are of similar size [11]. In contrast, the collections we seek to serve are often specialised and vary radically in size. We consider artificially created TREC sub-collections of generalist material (e.g. Wall Street Journal) to be a poor substitute for such targets.

Repositories and Queries We currently harvest 16 collections, ranging from 16 to over 800,000 documents each. The collection coverage is varied: some specialise in one subject (e.g. computer science); others address a range of subjects.

Our set of 50 test queries range from one to ten terms in length, with an average of four. Though short, it is not unreasonable: Silverstein [12] has shown that web queries often consist of only three terms or less. Some queries (built from document titles) target specific collections. Others consist of general descriptions of a wide subject area, known to be present in multiple collections.

Optimal Rank Generation As presented in Section 3.1, we have developed a new optimal ranking (F-Score based ranking), over which we will evaluate the rankings produced by the various collection selection algorithms. However, the FsBR requires document relevance judgements for each query; a resource we lack due to our use of real collections. As such, we utilise document ranking algorithms to generate surrogate relevance judgements for each test query.

We use the Apache Lucene[2] search engine library to build a document index from our harvested metadata. For each query we rank the documents using three

[1] http://openarchives.org/OAI/openarchivesprotocol.html
[2] http://lucene.apache.org/

algorithms: $tf.idf$, BM25 and the Lucene search algorithm; each returning the top 1000 relevant documents (around 0.09% of currently harvested documents). A list of pseudo-relevant documents is generated by taking the intersect of the relevant documents from the three algorithms. Thus, a document is relevant if all three algorithms agree it is relevant. From this list we can determine the number of relevant documents in each collection, and thus calculate their F-scores.

3.3 Doddle Algorithm

Our algorithm is inspired by criteria for highly ranked collections, presented in [16]. They state that a collection should be ranked highly if for each query term:

1. The term occurs in the collection;
2. The term is common in the collection (relative to the other collections);
3. The collection contains a relatively high proportion of documents with the term; and
4. There are likely to be documents in the collection in which the term is relatively frequent [16].

The Doddle algorithm ranks collections in decreasing order of *merit*. For a given query q, the merit associated with collection c is calculated by:

$$merit(q, c) = \sum_{t \in q} f_{q,t} \times (RC_{t,c} + RP_{t,c} + RF_{t,c})$$

where $f_{q,t}$ is the number of occurrences of term t in the query. The three components of the algorithm are described below:

Relative Commonness:	**Relative Proportion:**	**Relative Frequency:**
(deriv. from 2)	(deriv. from 3)	(deriv. from 4)
$RC_{t,c} = \frac{C_{t,c}}{\sum_{i=1}^{\lvert C \rvert} C_{t,i}}$	$RP_{t,c} = \frac{P_{t,c}}{\sum_{i=1}^{\lvert C \rvert} P_{t,i}}$	$RF_{t,c} = \frac{F_{t,c}}{\sum_{i=1}^{\lvert C \rvert} F_{t,i}}$

where:

$C_{t,c} = \frac{f_{c,t}}{tokens_c}$ (commonness of term t in collection c);
$P_{t,c} = \frac{df_{c,t}}{docs_c}$ (proportion of documents in collection c containing term t);
$F_{t,c} = \frac{f_{c,t}}{df_{c,t}}$ (average occurrences of term t in documents in collection c);
$f_{c,t}$ is the number of occurrences of term t in collection c;
$tokens_c$ is the total number of terms in collection c;
$df_{c,t}$ is the number of documents in collection c containing term t; and
$docs_c$ is the total number of documents in collection c.

4 Preliminary Results

Our initial experiment examines the suitability of the existing collection selection algorithms for our search task, and compares their performance to that of our Doddle algorithm.

Table 1 shows the results of the scenario and optimal performance tests (see Section 3.1 for details) conducted on the algorithms described in Section 2.1. The 'Scenarios' columns report which algorithms produced the correct ranking for each scenario[3]. Both CORI and bGlOSS show promise, as they succeeded for all seven scenarios. Inner Product and CVV failed on one and two scenarios respectively, suggesting they are less suited to our search task. The AvICTF algorithm performs very poorly, failing on five scenarios; two of which (one and four) are the most clear cut. This performance can be explained by the intended use for the algorithm: evaluating the quality of results at a single search engine. AvICTF tended to rank collection C_C first, which features very few occurrences of the query terms: these have high discriminating power, a property AvICTF deems desirable.

Algorithms	Scenarios							Spearman Rank Correlations	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**Titles**	**Titles & Descriptions**
AvICTF	✗	✗	✗	✗	✓	✓	✗	-0.537	-0.684
CVV	✓	✓	✓	✓	✓	✗	✗	-0.035	-0.179
Inner Product	✓	✓	✓	✓	✓	✓	✗	0.037	0.007
bGlOSS	✓	✓	✓	✓	✓	✓	✓	0.149	0.153
CORI	✓	✓	✓	✓	✓	✓	✓	0.226	0.106

Table 1. Algorithm performances during scenario and optimal performance tests.

The right columns of the table show the correlations between the algorithm ranking and the FsBR optimal (averaged over all 50 queries). To test how much data is required for an effective collection ranking, we compare the effect of using title metadata only, against both title and description metadata. We observe that in general, using only titles gives better correlations between the algorithm rankings and the optimal. However, the performance of all five algorithms was consistently poor. Although Inner Product, bGlOSS and CORI achieved positive correlations, none of the values were statistically significant (over an average of eight collection results), suggesting they are not suited to our interpretation of collection selection. As such, we may take bGlOSS and CORI as baselines for comparison, on which we should aim to improve.

Following the evaluation of the existing algorithms, we tested the Doddle algorithm using our two evaluation techniques. It was found to have encouraging performance; producing the correct rankings in all seven scenario tests. During

[3] The ✓ symbol represents a correct ranking; ✗ represents an incorrect ranking.

the optimal performance tests, the algorithm significantly out-performed CORI. In Figures 1 and 2 we show the Spearman rank correlation values both algorithms achieved against the optimal, over each query, and for both term indexes. We observe that Doddle frequently has higher correlation with the optimal: 0.75 or above on 22 queries (44%) executed over the title term index, whereas CORI achieved only 10 (20%) queries over this value. For queries executed over the title and description term index, Doddle had 16 (32%) with 0.75 correlation or above, against CORI's 6 (12%). These values suggest that the title term index enables a more accurate collection ranking. One explanation for this is that there is more noise in the 'description' metadata.

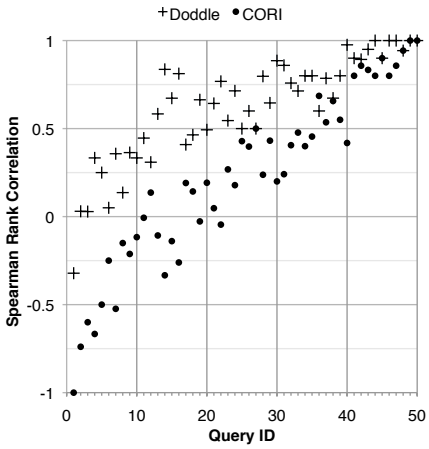

Fig. 1. Spearman rank correlations for queries over title terms.

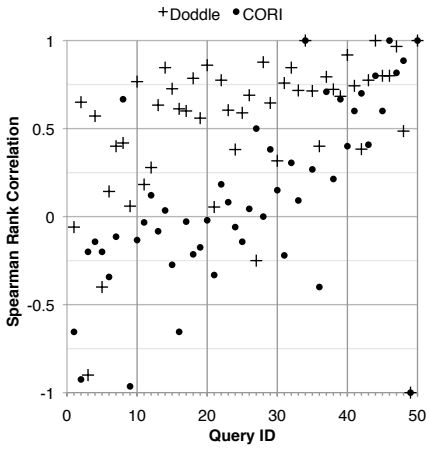

Fig. 2. Spearman rank correlations for queries over title and description terms.

Calculating the average correlation between Doddle and the optimal over the 50 queries, gives values of 0.624 over the title term index, and 0.518 over the title and description term index. For an average of eight collection results per query, these results are below the 5% significance level. Therefore, although the Doddle algorithm is more effective than CORI, there is still considerable room for improvement.

5 Future Work

This research is concerned with supporting the user in identifying digital library collections that will satisfy their current and future information needs. To this end, we have developed a methodology (based in part on the current collection selection evaluation method) and apparatus to support the evaluation of algorithms for this task. We have made progress in proving our hypotheses (specified

in Section 1): our initial experiment investigated the performance of existing collection selection algorithms, with the results suggesting they are ill-suited to our task. As such, we have begun development of a new algorithm. Whilst it has shown improvement over the existing algorithms, its correlation with an optimal ranking is not yet sufficient.

Our future work in this area will strive to refine both our evaluation methodology and collection selection algorithm. We see both tasks as iterative processes: making adjustments, testing and evaluating the adjustments, and thus making further alterations where required.

Improvements to our evaluation methodology will include a number of steps. The optimal performance test (Section 3.1) is a vital part of our evaluation process. As such, our first priority is to ensure that our optimal ranking produces the most sensible ranking of collections.

We recognise that our use of hand-built queries may be considered a weakness in our experimental methodology. Therefore, to further strengthen our evaluation process we will investigate alternative techniques for creating test queries, such as using query logs or automatic query generation.

Our initial experiments utilised only Spearman rank correlation to examine how well the algorithms estimated the optimal ranking. Before conducting future experiments we will expand the Doddle tool set to include algorithm evaluation using the other metrics mentioned in Section 2.2. We will also provide functionality to evaluate algorithms in terms of a *baseline* ranking.

Following these enhancements to our evaluation process, we will focus on improving the correlation between the Doddle algorithm and the optimal ranking. This will require careful analysis of the results, with the aim of identifying patterns in the queries where the algorithm performs poorly. It is hoped that these patterns will indicate suitable adjustments to the Doddle algorithm.

Once we are satisfied with our evaluation techniques, and have achieved a suitable performance from our algorithm, we will finally investigate which metadata (titles only or titles and descriptions) results in the most accurate collection ranking.

6 Feedback Issues

Our evaluation method is a key part of our research, and it is essential for the credibility of our results that we get it right. As such during the doctoral consortium, feedback on how to strengthen the evaluation technique would be appreciated. We would also welcome suggestions on alternative strategies for building test queries, and any other comments on the work as a whole.

Acknowledgements. Helen Dodd is supported by an EPSRC Doctoral Training Grant.

References

1. Buchanan, G., Cunningham, S.J., Blandford, A., Rimmer, J., Warwick, C.: Information seeking by humanities scholars. In: Proc. ECDL. LNCS, vol. 3652, pp. 218–229. Springer (2005)
2. Callan, J.P., Lu, Z., Croft, W.B.: Searching distributed collections with inference networks. In: Proc. SIGIR. pp. 21–28. ACM Press (1995)
3. Dodd, H., Buchanan, G., Jones, M.: A new perspective on collection selection. In: Proc. ECDL. LNCS, Springer (2010), to appear.
4. D'Souza, D.J., Zobel, J., Thom, J.A.: Is CORI effective for collection selection? An exploration of parameters, queries, and data. In: Proc. ADCS. pp. 41–46 (2004)
5. French, J.C., Powell, A.L.: Metrics for evaluating database selection techniques. World Wide Web 3(3), 153–163 (2000)
6. Fuhr, N.: A decision-theoretic approach to database selection in networked IR. ACM Trans. Inf. Syst. 17(3), 229–249 (1999)
7. Gravano, L., García-Molina, H., Tomasic, A.: The effectiveness of GlOSS for the text database discovery problem. In: Proc. SIGMOD. pp. 126–137. ACM Press (1994)
8. Gravano, L., García-Molina, H., Tomasic, A.: GlOSS: text-source discovery over the internet. ACM Trans. Database Syst. 24(2), 229–264 (1999)
9. He, B., Ounis, I.: Query performance prediction. Inf. Syst. 31(7), 585–594 (2006)
10. Meng, W., Yu, C., Liu, K.L.: Building efficient and effective metasearch engines. ACM Comput. Surv. 34(1), 48–89 (2002)
11. Powell, A.L., French, J.C.: Comparing the performance of collection selection algorithms. ACM Trans. Inf. Syst. 21(4), 412–456 (2003)
12. Silverstein, C., Marais, H., Henzinger, M., Moricz, M.: Analysis of a very large web search engine query log. SIGIR Forum 33(1), 6–12 (1999)
13. Srinivasa, R., Phan, T., Mohanraf, N., Powell, A.L., French, J.: Database selection using document and collection term frequencies. Tech. Rep. CS-2000-32, University of Virginia (May 2000)
14. Yuwono, B., Lee, D.L.: Server ranking for distributed text retrieval systems on the internet. In: Proc. DASFAA. pp. 41–50. World Scientific Press (1997)
15. Zhao, Y., Scholer, F., Tsegay, Y.: Effective pre-retrieval query performance prediction using similarity and variability evidence. In: Proc. ECIR. LNCS, vol. 4956, pp. 52–64. Springer (2008)
16. Zobel, J.: Collection selection via lexicon inspection. In: Proc. ADCS. pp. 74–80 (1997)