

A New Perspective on Collection Selection

Helen Dodd¹, George Buchanan², and Matt Jones¹

¹ Department of Computer Science, Swansea University, United Kingdom

² Centre for HCI Design, City University, London, United Kingdom

Abstract. Collection selection is traditionally a sub-problem of meta-search, and identifies collections most likely to contain relevant documents. However, we propose to treat collection selection as an independent search task with the goal of identifying collections that are relevant as a whole; so the user may return to them to serve future (related) information needs. Using a new methodology and framework we evaluate the suitability of existing collection selection algorithms for this search task, compared with a new algorithm designed specifically for the task.

Keywords: Collection selection, database selection, collection ranking.

1 Introduction

Consider a scenario where a user wants to locate authoritative collections (e.g. digital libraries) on a particular topic, to fulfil both current and *future* information needs. A technique to identify collections with a degree of relevance to a query is *collection selection*; a sub-problem of metasearch. It supports *document retrieval* by choosing a subset of collections most likely to contain relevant documents. The query is dispatched to these collections, and the results merged to form a list of relevant documents [5]. We propose to treat collection selection as an independent search task. Here the goal is not to find individual documents from multiple collections, but to identify individual collections containing a high *proportion* and *quantity* of relevant documents: collections *about* the query.

We present a methodology and framework for the evaluation of algorithms over our interpretation of collection selection. These techniques are used to evaluate the performance of a new algorithm, designed specifically for our task; and to examine the suitability of existing collection selection algorithms.

2 Our Evaluation Method

To test the suitability and performance of algorithms for our retrieval task we use *scenario* and *optimal performance* tests. The scenario tests use controlled data to scrutinise the behaviour of algorithms over clear cases: algorithms producing incorrect rankings will not suit our needs. Each of our seven scenarios models three collections, one of which is the clear winner, another the clear loser. Different attributes (size, term frequency, quantity/proportion of relevant documents) of the collections are varied in each scenario.

The optimal performance test surveys how well algorithms estimate³ an *optimal* ranking. Traditionally [2] the optimal orders collections by the number of relevant documents they contain. However, we propose an optimal more representative of our search task; where suitable collections contain a high number and proportion of relevant documents. We represent this with two metrics:

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

where RS_c is the *share* and RP_c the *proportion* of relevant documents in a collection c . We order collections by decreasing harmonic mean (F-score) of RS_c and RP_c : the F-score Based Ranking (FsBR). We use the Spearman rank correlation coefficient to determine how well algorithm rankings estimate the optimal.

3 Our Algorithm

Our algorithm (called *Doddle*) is inspired by criteria for highly ranked collections [8]: if each query term is *common* and occurs *frequently* in a high *proportion* of documents within a collection (relative to other collections), the collection should be highly ranked. Doddle ranks collections in decreasing order of *merit*. For a given query q , the merit associated with collection c is calculated by:

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

where $f_{q,t}$ is the number of occurrences of term t in the query and:

$$RC_{t,c} = \frac{C_{t,c}}{\sum_{i=1}^{|C|} C_{t,i}} \quad RP_{t,c} = \frac{P_{t,c}}{\sum_{i=1}^{|C|} P_{t,i}} \quad RF_{t,c} = \frac{F_{t,c}}{\sum_{i=1}^{|C|} F_{t,i}}$$

(Relative Commonness) (Relative Proportion) (Relative Frequency)

where:

$$C_{t,c} = \frac{f_{c,t}}{\text{tokens}_c} \text{ (commonness of term } t \text{ in collection } c);$$

$$P_{t,c} = \frac{df_{c,t}}{\text{docs}_c} \text{ (proportion of documents in collection } c \text{ containing term } t);$$

$$F_{t,c} = \frac{f_{c,t}}{df_{c,t}} \text{ (average occurrences of term } t \text{ in documents in collection } c);$$

$f_{c,t}$ 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 Apparatus

We support the evaluation of algorithms with a set of applications that enable the management of collection data, creation of scenarios, and the execution of tests and the display of their results.

³ Methods indicating how well an algorithm estimates an optimal include: Mean-squared error; Spearman rank correlation coefficient; and a recall-based metric [2].

Our *optimal performance tests* use 16 collections, ranging from 16 to 800,000 documents in size. Their coverage is varied: some specialise in one subject, others address a range of subjects. They are real collections, harvested using OAI-PMH⁴. We harvest the Title and Description fields in Dublin Core format and create two indexes from the data: “title only” and “title and description”.

Previous studies have created artificial collections from TREC data (divided by source and date, or size [6]). Such collections are often of generalist material, and are thus a poor substitute for those we aim to serve: specialised and of varying size. However, using real collections leaves us without document relevance judgements, required by FsBR in the optimal performance tests. We therefore generate sudo-relevance judgements using document ranking algorithms. The Apache Lucene⁵ library is used to create a document index from the harvested metadata. For each query, the documents are ranked by *tf.idf*, BM25 and the Lucene search algorithm. Documents that all three algorithms agree are relevant are appended to a list of “relevant” documents. From this we determine the number of relevant documents in each collection, and calculate their F-scores.

We use a test set of 50 queries (1-10 terms long). Some queries target specific collections, others describe wide subject areas, present in multiple collections.

5 Experimental Results

Our initial experiments use our scenario and optimal performance tests to survey the suitability of existing algorithms for our task, and compare their performance to Doodle. We investigate algorithms previously shown to be effective: CORI [1] (often used as a benchmark); bGLOSS [3]; Cue Validity Variance (CVV) [7]; and Inner Product [8]. We also consider Average Inverse Collection Term Frequency (AvICTF) [4], a *query performance predictor* which will produce rankings based on the predicted quality of results (were each collection searched in isolation).

In the scenario tests, Inner Product, CVV and AvICTF were found to be ill-suited to our search task; failing on one, two and five scenarios respectively. Specifically, AvICTF frequently favoured collections with the least query term occurrences. CORI and bGLOSS succeeded for all seven scenarios, suggesting they may be suitable baselines for comparison with our own algorithm; which also produced correct rankings in all scenarios.

Table 1 shows the average correlations between the algorithm rankings and the optimal. We also compare the effect of using only title metadata, versus both titles and descriptions. Our algorithm produces rankings with a much higher correlation to the optimal ranking than any of the existing algorithms. However, for an average of eight collection results per query, the correlation is below the 5% significance level: there is still considerable room for improvement.

Most algorithms produced a better correlation with the optimal when queries were executed over the “title only” term index. One explanation may be that the description metadata has more noise, however further investigation is required.

⁴ <http://openarchives.org/OAI/openarchivesprotocol.html>

⁵ <http://lucene.apache.org/>

Algorithms	Titles	Titles and 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
Doddle	0.624	0.518

Table 1. Average Spearman rank correlation coefficients for each algorithm.

6 Conclusions and Future Work

We have presented a new interpretation of collection selection: to treat it as an independent search task, with the goal of identifying quality collections that will satisfy a user’s current and future information needs. The paper reported our methodology to evaluate algorithms for this task. With this, we investigated the performance of existing collection selection algorithms, compared to that of our own algorithm. Our algorithm significantly outperformed existing algorithms; however, its correlation with an optimal ranking was still not satisfactory.

Our future work will iteratively improve our evaluation methodology and the selection algorithm. This will include the refinement of the scenario and optimal performance tests. In particular, we aim to ensure our optimal ranking produces the most sensible ordering of collections. Future experiments will evaluate algorithms in terms of a baseline ranking, and use additional evaluation metrics.

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

References

1. Callan, J.P., Lu, Z., Croft, W.B.: Searching distributed collections with inference networks. In: Proc. SIGIR. pp. 21–28. ACM Press (1995)
2. French, J.C., Powell, A.L.: Metrics for evaluating database selection techniques. *World Wide Web* 3(3), 153–163 (2000)
3. 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)
4. He, B., Ounis, I.: Query performance prediction. *Inf. Syst.* 31(7), 585–594 (2006)
5. Meng, W., Yu, C., Liu, K.L.: Building efficient and effective metasearch engines. *ACM Comput. Surv.* 34(1), 48–89 (2002)
6. Powell, A.L., French, J.C.: Comparing the performance of collection selection algorithms. *ACM Trans. Inf. Syst.* 21(4), 412–456 (2003)
7. 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)
8. Zobel, J.: Collection selection via lexicon inspection. In: Proc. ADCS. pp. 74–80 (1997)