Is CORI Effective for Collection Selection? An Exploration of Parameters, Queries, and Data

Daryl D'Souza Justin Zobel James A. Thom
School of Computer Science & Information Technology
RMIT University, Melbourne 3001, Australia
{djds,jz,jat}@cs.rmit.edu.au

Abstract *In distributed information retrieval, a wide range of techniques have been proposed for choosing collections to interrogate. Many of these collection-selection techniques are based on ranking the lexicons; of these, arguably the best known is the CORI collection ranking metric, which includes several parameters that, in principle, should be tuned for different data sets. However, parameters chosen in early work on CORI have been used without alteration in almost all subsequent work, despite drastic differences in the data collections. We have explored the behaviour of CORI for a range of data sets and parameter values. It appears that parameters cannot reliably be chosen for CORI: not only do the optimal choices vary between data sets, but they also vary between query types and, indeed, vary wildly within query sets. Coupled with the observation that even CORI with optimal parameters is usually less effective than other methods, we conclude that the use of CORI as a benchmark collection selection method is inappropriate.*

Keywords Lexicon indexing, distributed retrieval, information retrieval.

1 Introduction

In distributed information retrieval, the search process involves passing the query to each of a set of search servers, then collating their responses. Each such server indexes a collection of documents. The cost of search can be reduced by only passing the query to a limited number of servers, giving rise to the *collection selection* problem: identification of those collections most likely to contain answers.

A method of collection selection that has been widely described in the research literature is to use information about each collection's lexicon (Callan, Lu & Croft 1995, Craswell, Bailey & Hawking 2000, French, Powell, Callan, Viles, Emmitt, Prey & Mou 1999, Gravano & Garcia-Molína 1995, Hawking & Thistlewaite 1999, Meng, Yu & Liu 2002, Yuwono & Lee 1997, Zobel 1997). In a common approach, the central service maintains a copy of the complete lexicon of each collection, which should be much smaller than indexes for the collections, or the text of the collections themselves. These lexicons can then be cheaply compared to the query to establish which are the most promising.

Proceedings of the 9th Australasian Document Computing Symposium, Melbourne, Australia, December 13, 2004.

A range of query-to-lexicon similarity measures have been proposed. Of these, arguably the best known is CORI, first described by Callan et al. (1995), and used in their subsequent work (Callan 2000, French et al. 1999, Larkey, Connell & Callan 2000). It has been used in many papers as a standard (Abbaci, Savoy & Beigbeder 2002, Callan & M.Connell 2001, Callan 2000, Lu & McKinley 1999, Powell, French, Callan, Connell & Viles 2000, Rasolofo, Abbaci & Savoy 2001, Si & Callan 2002), and even some recent papers report experiments showing it to be an effective collection-selection metric (Conrad, Guo, Jackson & Meziou 2002, French et al. 1999, Larkey et al. 2000, Si & Callan 2003). While this verdict is not universal, this continued investigation of CORI demonstrates that many researchers view it as a key approach.

In recent work, we were surprised by the relatively poor behaviour of CORI on some datasets, in the worst cases achieving only around half the effectiveness of other methods (D'Souza, Thom & Zobel 2004). We investigated the results to identify the issues—was our implementation at fault, for example? What we discovered was rather more serious. CORI is derived from a theoretical argument, giving a formulation in which key parameters are undetermined. In an early CORI paper, values for these parameters were chosen by exploration on a particular collection (Callan et al. 1995). In all subsequent papers the same parameter values have been used, with no reported attempt to reinvestigate them.

We searched for the best values for the CORI parameters over several sets of collections and queries, and found that in many cases the standard parameters give significantly inferior results than those observed with other parameter choices. For comparison, we show results on the same collections achieved by methods discussed by Zobel (1997); in 13 of 21 cases, the simple (and unparameterised) Inner product is superior to standard CORI, while in 15 of 21 cases the CORI is worse than Highsim, the other method tested. In 9 of the 21 cases, standard CORI is worse than the baseline of sized-based ranking (SBR) of simply selecting the largest collections.

Not only does the optimal choice of CORI parameter values vary dramatically from test set to test set, but it varies dramatically between query sets on the same data and between queries of the same type on the same data. Moreover, other issues are suggested by our investigation. While CORI has performed well in some other experiments, it appears that is because the test data con-

sisted of a small number of similarly-sized collections of unrelated documents—an environment that does not allow much discrimination between methods. Taking these problems together, it is far from clear that CORI is a wise choice of collection-selection metric.

2 CORI

In the collection-selection problem, it is assumed that the set C of collections has N members, and collection $c \in C$ contains f_c documents. Many scoring mechanisms used for selecting collections are rather like conventional similarity measurement, in that each collection is treated as a bag of words, just as a document is a bag of words in document retrieval. Therefore similar statistics are used, in particular $f_{c,t}$, the number of documents containing term t in collection c, and f_t, the number of collections containing t. Collections that score the highest are assumed to be the most likely to contain documents that are relevant to the query.

CORI is based on Bayesian inference networks. In CORI the similarities between a user query and a set of known document collections is computed, in order to rank the collections. The query is then submitted to each selected (highly ranked) collection to retrieve its set of top ranked documents; these separate document sets are then merged.

The similarity of a query to a given collection is the sum of the belief probabilities of the query terms appearing in the collection. The CORI similarity between a query q and collection c can be computed as

$$\mathit{CORI}(q,c) = \frac{\sum_{t \in q\&c} (d_b + (1-d_b) \cdot T_{c,t} \cdot I_{c,t})}{|q|}$$

where d_b, the minimum belief component, is set to 0.4, $T_{c,t}$ is the weight of the term in the collection, $I_{c,t}$ is the inverse collection frequency, and $|q|$ is the number of distinct terms in the query. The value $|q|$ can be ignored as it is constant for a given query. The inverse collection frequency $I_{c,t}$ can be computed as

$$I_{c,t} = \frac{\log((N+0.5)/f_t)}{\log(N+1.0)} \quad (1)$$

In an early version of CORI (Callan et al. 1995), the weight $T_{c,t}$ is computed as

$$T_{c,t} = d_t + (1-d_t) \cdot \frac{\log(f_{c,t}+0.5)}{\log(max_c+1.0)}$$

where d_t is the minimum term frequency component, set to 0.4 in earlier experiments (Allan, Ballesteros, Callan, Croft & Lu 1995, Callan et al. 1995), and max_c is the number of documents containing the most frequent term in collection c.

A suggested improvement to $T_{c,t}$ is to scale $f_{c,t}$ by adding a constant K (Callan et al. 1995). When ranking collections, it was argued, it is better to make K sensitive to the number of documents (as opposed to percentage of documents) on a topic. Furthermore, it was proposed that K should be large, because the $f_{c,t}$ values will generally be large. The computation of $T_{c,t}$ is modified to

$$T_{c,t} = d_t + (1-d_t) \cdot \frac{f_{c,t}}{f_{c,t}+K} \quad (2)$$

where K is computed as

$$K = k \cdot \big((1-b) + b \cdot (F_c/\bar{F}_c)\big) \quad (3)$$

where $\bar{F}_c = \sum_{c \in C} F_c/N$, and k and b are parameters. The parameter k controls the magnitude of K, while varying b from 0 to 1 increases the sensitivity of K to the size of the collection. The value F_c is the "number of word (term) occurrences" in c (French et al. 1999).

In the initial description of CORI, experiments were used to identify suitable k and b values (Allan et al. 1995, Callan et al. 1995). The first TREC CD (disk volume 1) was broken into seven collections, varying in size from a few million to a few tens of millions of words, and was tested with queries 51–100. The space of k and b values was searched using the relevance judgements for this data, giving the values $k = 200$ and $b = 0.75$. This yields

$$T_{c,t} = d_t + (1-d_t) \cdot \frac{f_{c,t}}{f_{c,t} + 50 + 150 \cdot F_c/\bar{F}_c}$$

$$\mathit{CORI}(q,c) = \frac{\sum_{t \in q\&c} (d_b + (1-d_b) \cdot T_{c,t} \cdot I_{c,t})}{|q|} \quad (4)$$

In some experiments (Callan 2000, French et al. 1999, Larkey et al. 2000), d_t is dropped; that is, the previously used default (Callan et al. 1995) of $d_t = 0.4$ is replaced by $d_t = 0$.

In the experiments reported in this paper, we explore the choice of values for the parameters d_t, k, and b. We use CORI as in Equation 4, and $I_{c,t}$, $T_{c,t}$, and K as in Equations 1, 2 and 3 respectively. In almost every paper that uses CORI, the values used for the key parameters are $k = 200$ and $b = 0.75$. The only exceptions of which we are aware are the work of Conrad et al. (2002), where it is reported (without discussion of how the parameters were explored) that $k = 300$ and $b = 0.6$ are superior; and of Lu & McKinley (1999), where a small number of combinations are explored in the context of replication.

As a thought experiment, it is interesting to examine the expected behaviour of CORI in different environments, using default parameters. Values of N, f_c, and F_c are shown for our test data (discussed in the next section) in Table 1. Consider now the collection DATELINE-M. The ratio $F_c/\bar{F}_c$ varies from 0.00039 to 119.62. For a rare term with $f_{c,t} = 1$, possible $T_{c,t}$ values range from 0.020 down to 0.000056. For $f_{c,t} = 100$, the value of $T_{c,t}$ for the largest value of F_c rises to 0.0056. Even if $f_{c,t} = 1000$, the value rises to only 0.053. Thus a large collection can only be selected if it contains a large number of documents with one of the query terms—it is unlikely that a collection with a small number of relevant documents would be highly ranked. Conversely, a small collection with a couple of

Table 1: *Data set summaries, showing the number of collections N and the distribution of f_c and F_c values.*

	N	f_c			F_c		
		min	avg	max	min	avg	max
ZDISK2	43	1,642	5,377	7,888	1,317,038	1,716,025	1,948,747
ZDISK3	91	14	3,696	22,853	996	1,002,430	19,494,646
ORIGINAL17	17	6,711	63,422	226,087	2,898,248	15,783,397	29,996,344
SYM236	236	1	2,928	8,302	500	943,716	2,653,311
UDC236	236	2,891	2,928	3,356	588,842	943,716	8,863,449
BYLINE-M	2,239	1	39	6,440	21	13,173	1,848,436
BYLINE-C	2,239	1	39	6,440	18	13,173	2,139,015
BYLINE-R	2,239	1	39	6,440	71	13,173	2,141,808
DATELINE-M	530	1	289	30,507	29	75,083	8,981,080
DATELINE-C	530	1	289	30,507	61	75,083	7,819,445
DATELINE-R	530	1	289	30,507	23	75,083	7,948,978

occurrences of any query term is automatically highly ranked. Experimentally, we found that CORI rarely ranks large collections highly, even though they are often the best source of relevant documents.

In a recent paper, Si & Callan (2003) explore the limitations of CORI when collection size varies, and found the same defect. They propose modification to CORI based on estimated database size to compensate for this effect but they do not address the difficult issue of parameter choice. We plan to test this compensation in future work, but it is not clear that the positive results would be observed in the collections we use, where the variation in size and number of collections is in some cases much greater.

3 Test data

We use a range of test sets in our experiments. The first and second sets are contents of TREC disks 2 and 3. They are denoted ZDISK2 and ZDISK3, divided into 43 and 91 collections, respectively (Zobel 1997). In the former of these, each of the 43 collections is of similar size. In the latter, the divisions between the 91 collections were chosen at random, and the sizes vary dramatically.

The third data set used is ORIGINAL17, the contents of TREC disks 1 to 3 divided into seventeen collections as in the original work of Callan, Lu, and Croft (they used seven of these collections to determine the default CORI parameters). Collection sizes vary from 6,711 documents to 226,087 documents. As can be seen, this set is very different to the others.

The fourth data set, denoted SYM236 was developed by French, Powell, Viles, Emmitt & Prey (1998) to explore selection in larger databases (with at least 100 collections), and was a partitioning of TREC data based on source, year and month boundaries. The fifth data set, denoted UDC236 is reorientation of this same set of documents partitioned on the basis of approximately equi-sized collections (Powell et al. 2000).

The last six were derived from the Associated Press data on TREC CDs 1 and 2 (Harman 1995), and have been used by us in other work to explore the impact of different ways of classifying documents (D'Souza et al. 2004). The first of these is BYLINE-M, where the data was divided into 2239 collections according to the <BYLINE> field. Documents without a byline were omitted. Collection sizes were highly skew—most had only one or two documents, 150 or so had between 100 and 700 documents, and one had 6440 documents. We hypothesised that such a breakdown might reflect how documents were created and stored in a workplace such as a news provider, and thus provides a realistic real-world test of distributed document retrieval.

The second of these sets was BYLINE-C; this was a chronological breakdown of exactly the same set of documents into 2239 collections of exactly the same distribution of sizes. The third of these sets was BYLINE-R; a random breakdown of the same set of documents into 2239 collections of exactly the same distribution of sizes.

The fourth of these AP-sourced sets was DATELINE-M, where documents were classified by the <DATELINE> field. This yielded 153,020 documents in 530 collections; the sizes were again skew. The second last of these sets was DATELINE-C, a chronological breakdown of the DATELINE documents into 530 collections with the same size distribution. The last of these sets was DATELINE-R, a random breakdown of the DATELINE documents the same size distribution.

The TREC queries include a short form, or heading, and a longer exposition. This allows each query set to be used twice, in SHORT or LONG form. For ZDISK3, only short queries are available.

4 Experiments

A standard way to measure the performance of collection-selection metrics is to count the number of relevant documents in the highly-ranked collections. For each query, the number of documents that have been judged relevant is known, thus each collection makes a known contribution to the recall for that query. For example, suppose that for a given query there are 200 known relevant documents, and the three collections ranked highest have 16, 2, and 10 relevant documents respectively. Then the recall

Table 2: *Best CORI parameter combinations and recall@10 figures for each test set and query set, compared with baselines SBR, RBR and with standard CORI,* Highsim*, and Inner product.*

		Best (k, b, d_t)	Baselines SBR, RBR	Best CORI	Std. CORI	High-Sim	Inner product
ZDISK2	LONG	200.00, 0.8, 0.4	37.1, 77.1	57.0	**56.4**	54.2	54.9
ZDISK2	SHORT	21.54, 0.4, 0.8	37.1, 77.1	55.1	51.6	**54.0**	53.1
ZDISK3	SHORT	215.44, 0.0, 0.6	36.2, 78.8	51.6	39.6	35.0	**51.1**
ORIGINAL17	LONG	215.44, 1.0, 0.4	78.2, 99.3	93.5	**91.8**	83.6	82.9
ORIGINAL17	SHORT	464.16, 0.75, 0.6	78.2, 99.3	92.5	**91.8**	86.2	85.4
SYM236	LONG	1000.00, 0.6, 0.8	11.0, 47.1	25.7	**25.3**	23.6	22.7
SYM236	SHORT	464.16, 0.2, 0.6	11.0, 47.1	23.5	**22.2**	**22.2**	21.5
UDC236	LONG	464.16, 0.8, 0.2	1.8, 36.4	12.6	11.7	**15.1**	9.5
UDC236	SHORT	100.00, 0.4, 0.0	1.8, 36.4	13.4	12.6	**14.4**	11.3
BYLINE-M	LONG	2.15, 0.6, 0.2	11.2, 75.8	43.1	39.2	**39.8**	35.6
BYLINE-M	SHORT	21.54, 0.2, 0.6	11.2, 75.8	37.3	27.9	**39.8**	35.4
BYLINE-C	LONG	4.46, 0.2, 0.8	13.6, 52.9	18.0	7.9	**18.1**	16.7
BYLINE-C	SHORT	10.00, 0.0, 0.0	13.6, 52.9	17.6	4.5	**20.3**	17.6
BYLINE-R	LONG	10.00, 0.2, 0.4	12.8, 49.0	16.9	6.5	**16.5**	15.0
BYLINE-R	SHORT	1.00, 0.0, 0.2	12.8, 49.0	16.5	2.9	**19.0**	16.1
DATELINE-M	LONG	21.54, 0.4, 0.8	59.8, 89.7	70.6	60.0	**67.3**	66.4
DATELINE-M	SHORT	21.54, 0.0, 0.8	59.8, 89.7	69.0	34.1	**69.8**	68.0
DATELINE-C	LONG	21.54, 0.4, 0.8	48.5, 69.8	50.2	40.3	**49.8**	49.1
DATELINE-C	SHORT	21.54, 0.0, 0.8	48.5, 69.8	49.6	20.5	**49.7**	49.4
DATELINE-R	LONG	21.54, 0.4, 0.8	49.1, 65.6	49.6	46.8	**49.3**	49.2
DATELINE-R	SHORT	100.00, 0.0, 0.8	49.1, 65.6	49.3	24.8	**49.2**	49.0

available if the query is only passed to the top-ranked collection—that is, the recall@1—is 8%. The recall@3 is $(16 + 2 + 10)/200 = 14\%$.

Two benchmarks can be used to bracket possible performance. One is the "perfect" (or RBR, relevance-based ranking) score, if the collections are sorted by decreasing numbers of relevant documents. It is not possible to exceed this figure. The other is the "fixed" (or SBR, sized-based ranking) score, if the collections are sorted by decreasing size, on the simple heuristic that large collections should contain more relevant documents. Selection metrics that do no better than the fixed ordering are uninteresting. Plotting recall against the number of collections, it can be seen that a typical value such as recall@10 is a good indicator of overall performance.

In our experiments, we explored ranges of values for each of k, b, and d_t. The k values are large but not unbounded; we let k range through the geometric series given by progressively dividing 1000 by the value $\sqrt[3]{10} = 2.1544$, terminating at 0.21544, giving 13 values in total. In addition, we tested $k = 200$. We let b take the values 0.00, 0.20, 0.40, 0.60, 0.75, 0.80, and 1.00. We let d_t take the values 0.0, 0.2, 0.4, 0.6, 0.8, and 1.0. A value of $b = 1.0$ means that CORI simply counts the terms in common between query and collection; a value of $d_t = 1$ means CORI just sums up the $I_{c,t}$ terms. With inclusion of the values $k = 200$ and $b = 0.75$, the standard CORI ($d_t = 0.0$) is one of the values tested. In total $13 \times 7 \times 6 = 546$ combinations were explored for each set of queries and test set. We additionally tested varying d_b, but not as exhaustively.

In our first experiment, we set $b = 0.75$ and $d_t = 0.0$, and varied k. Results are shown in Figure 1. In this figure, the values shown are recall at 10 documents retrieved for each value of k and each collection and query set, where the recall values have been rescaled so that 0 is the recall for the "fixed" baseline and 100 is the recall for the "perfect" baseline. (The values of these baselines are shown in Table 2.)

As can be seen, the peak k figure varies wildly. Considering the SHORT queries shown in the left-hand-side graphs, the best value varies from 0.22 to 464.42, depending on the collection, and incorrect choice of k can seriously degrade precision: tuning on one data set does not give good results on the other data sets. This effect is even more pronounced for the LONG queries shown in the right-hand-side graphs, where the best k values range from 0.22 to 1000.00: the best k value for BYLINE-M reduces the score for ZDISK2 from 49 to 20.

More disturbingly, the best k value also varies between query sets on the same data. In the worst instance, DATELINE-M, the best value is 10.00 for SHORT but is 1000.00 for LONG. These queries were derived from the same topics. The parameters commonly used for CORI, which have been justified by performance on only one set of collections, are in some cases an extremely poor choice. Indeed, it is difficult to identify a good choice of parameters for a given set of documents and a query type. We counted the number of queries for which each of the tested k values is the best choice. For example, of the 100 SHORT queries on ZDISK2, there were eight for which $k = 0.22$ was best and 23 for which $k = 1000.00$ was best. Overall, the best choice of k for this data was 21.54 (see Table 2), despite

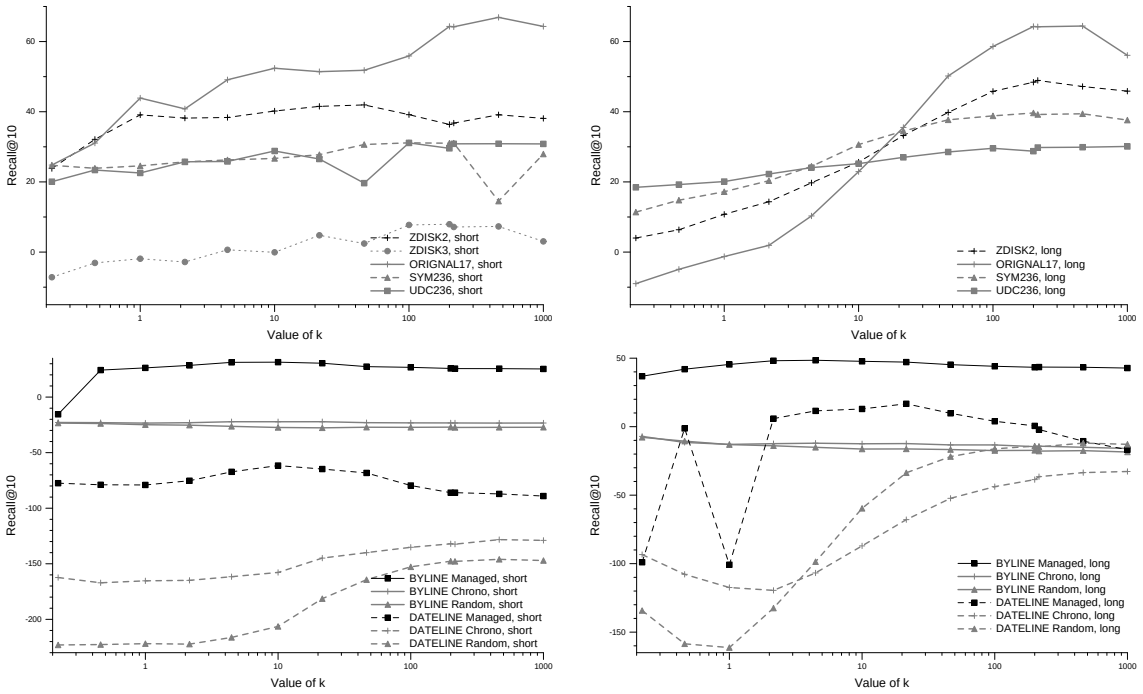

Figure 1: *Effect on recall@10 of varying k for fixed $b = 0.75$ and $d_t = 0.0$, on all collections. Left graphs: for* SHORT *queries. Right graphs: for* LONG *queries. In each case, the recall@10 has been rescaled so that the "fixed" SBR result is 0 and the "perfect" RBR result is 100.*

the fact that this was not the best choice for the great majority of queries.

The effect of varying b is similar to the effect of varying k, but not as extreme. Holding k constant at values such as 10 and 200, and varying b, we observed variations of up to about 15% on the rescaled recall values. These are not vast changes, but neither are they insignificant.

To identify the degree to which CORI performance varies for alternative choices of parameters, as discussed above we completely explored the $13 \times 7 \times 6$ space of parameter values to find the best combination in each case.

Results are in Table 2, which shows, for each test set and query set, the most successful combination of CORI parameters, the recall@10 achieved for this combination, and the recall@10 for the standard combination. Results for two other methods, Highsim and Inner product, are also shown; the figure in bold in each line is the best of standard CORI, Highsim, and Inner product.

We found, not surprisingly, that the poorest CORI results were observed with $d_t = 1.0$. However, for all other values of d_t, the value chosen had no discernible effect on performance. Similarly, we tested d_b, and found that it had little effect. For this reason we do not report changes in performance as a function of these two parameters; note that in many of the previous papers on CORI the parameter d_b is fixed at 0.4.

These results show the best b value varying from 0.0 to 1.0, and the best k value varying from 1.00 to 1000.00. The largest set of collections, the BYLINE data, showed the greatest improvement by tuning CORI and the greatest deviation from the original parameters. There is no consistent optimal combination of parameters for best CORI; indeed, no best CORI values coincide with the choice of $k = 200$ and $b = 0.75$ for standard CORI.

Even the best CORI results are often below the results observed with other lexicon-based methods, such as the Highsim method of Zobel (1997) or even the simple Inner product. Table 2 shows that these other ranking formulations often outperform CORI. Highsim is superior to CORI in 8 of the 21 cases, and superior to best CORI in 8 of the 21 cases; Inner product, although generally poorer than Highsim, is superior to CORI in 13 out of 21 cases. Only on the smallest set of collections, ORIGINAL17, is CORI the best method.

5 Conclusions

CORI has been used in a range of experiments in recent work, in some cases in comparisons with other collection-selection algorithms. In most of these experiments, the CORI formulation has used fixed values for parameters k (set to 200) and b (set to 0.75). These values were based on a set of seven collections extracted from TREC disk 1.

Experiments that used these recommended k and b values did so within a variety of experimental settings. We explored several variations with the aim of establishing optimal choices of k and b for variations in several factors. Our analysis of CORI, within this framework, showed that the greatest CORI effectiveness was for parameter values that did not coincide with

the usual choice. The experiments show that the CORI parameters k and b are highly sensitive to the variations in data sets, and that best k values are widely distributed even for a single data set and type of query. There is no obvious mechanism for setting the CORI parameters, and the use of standard CORI as a benchmark collection-selection method is not justified.

Acknowledgements

This work used computing facilities supported by the Australian Research Council.

References

Abbaci, F., Savoy, J. & Beigbeder, M. (2002), A methodology for collection selection in heterogeneous contexts, *in* 'Proc. IEEE Int. Conf. on Information Technology: Coding and Computing (ITCC'02)', Las Vegas, USA, pp. 529–535.

Allan, J., Ballesteros, L., Callan, J., Croft, W. B. & Lu, Z. (1995), Recent experiments with INQUERY, *in* D. Harman, ed., 'Proc. Fourth Text Retrieval Conf. (TREC-4)', National Institute of Standards and Technology Special Publication 500-236, Gaithersburg, Maryland, pp. 49–57.

Callan, J. & M.Connell (2001), 'Query-based sampling of text databases', *ACM Transactions on Information Systems* **19**(2), 97–130.

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', Kluwer Academic Publishers, pp. 127–150.

Callan, J. P., Lu, Z. & Croft, W. B. (1995), Searching distributed collections with inference networks, *in* E. A. Fox, P. Ingwersen & R. Fidel, eds, 'Proc. ACM SIGIR Int. Conf. on Research and Development in Information Retrieval', ACM, pp. 21–28.

Conrad, J., Guo, X., Jackson, P. & Meziou, M. (2002), Database selection using actual physical and acquired logical collection resources in a massive domain-specifi c operational environment, *in* 'Proc. 28th Int. Conf. on Very Large Data Bases (VLDB'02)', Hong Kong, China.

Craswell, N., Bailey, P. & Hawking, D. (2000), Server selection on the world wide web, *in* 'Proc. Fifth ACM Conf. on Digital Libraries', pp. 37–46.

D'Souza, D., Thom, J. A. & Zobel, J. (2004), 'Collection selection for managed distributed document databases', *Information Processing and Management* **40**(3), 527–546.

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, 'Proc. ACM SIGIR Int. Conf. on Research and Development in Information Retrieval', ACM, pp. 238–245.

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, 'Proc. ACM SIGIR Int. Conf. on Research and Development in Information Retrieval', ACM, pp. 121–129.

Gravano, L. & Garcia-Molína, H. (1995), Generalising GlOSS to vector-space databases and broker hierarchies, *in* 'Proc. Int. Conf. on Very Large Data Bases, September 11-15, Zurich, Switzerland', pp. 78–89.

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.

Larkey, L., Connell, M. & Callan, J. (2000), Collection selection and results merging with topically organized U.S. patents and TREC data, *in* 'Proc. Ninth ACM Int. Conf. on Information and Knowledge Management (CIKM'00)', pp. 282–289.

Lu, Z. & McKinley, K. (1999), Partial replica selection based on relevance for information retrieval, *in* 'Proc. ACM SIGIR Int. Conf. on Research and Development in Information Retrieval', ACM, Berkley, CA USA, pp. 97–104.

Meng, W., Yu, C. & Liu, K.-L. (2002), 'Building effi cient 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* 'Proc. ACM SIGIR Int. Conf. on Research and Development in Information Retrieval', ACM, pp. 232–239.

Rasolofo, Y., Abbaci, F. & Savoy, J. (2001), Approaches to collection selection and results merging for distributed information retrieval, *in* 'Proc. Tenth ACM Int. Conf. on Information and Knowledge Management, 2001 (CIKM'01)', Atlanta, USA, pp. 191–198.

Si, L. & Callan, J. (2002), Using sampled data and regression to merge search engine results, *in* 'Proc. ACM SIGIR Int. Conf. on Research and Development in Information Retrieval', ACM, Tempere, Finland, pp. 19–26.

Si, L. & Callan, J. (2003), Relevant document distribution estimation method for resource selection, *in* 'Proc. ACM SIGIR Int. Conf. on Research and Development in Information Retrieval', Toronto, Canada, pp. 298–305.

Yuwono, B. & Lee, D. L. (1997), Server ranking for distributed text retrieval systems on the internet, *in* 'Proc. 5th Int. Conf. on Database Systems for Advanced Applications (DASFAA'97)', Melbourne, Victoria, Australia, pp. 41–49.

Zobel, J. (1997), Collection selection via lexicon inspection, *in* 'Proc. 2nd Australian Document Computing Symposium (ADCS'97)', Melbourne, Victoria, Australia, pp. 74–80.