Reducing Storage Costs for Federated Search of Text Databases

Jie Lu
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
jielu@cs.cmu.edu

Jamie Callan
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
callan@cs.cmu.edu

Abstract

In environments containing many text search engines a federated search system provides people with a single point of access. When search engines are managed by independent organizations two key problems are discovering and representing the contents of each text database. Q*uery-based sampling* is a recent technique for discovering the contents of uncooperative databases so as to create database *resource descriptions* that support a variety of necessary capabilities. However, when the documents obtained by query-based sampling are very long, as is common in some government environments, disk storage costs can be surprisingly large. This paper investigates methods of pruning sampled documents to reduce storage costs. The experimental results demonstrate that disk storage costs can be reduced by 54-93% while causing only minor losses in federated search accuracy.

1. Introduction

More than a hundred thousand text databases ("search engines", "digital libraries") are available on the Internet today. These databases span a range of topics, languages, quality, and access characteristics, making it difficult for a user to know where to search for material on a specific topic. One solution is a single user interface that provides federated search of the available text databases. A person submits a query in the usual manner, a central site selects an appropriate set of databases ("resource selection"), submits the query to the selected databases, and merges the ranked lists of documents returned by each database into a single, integrated ranked list ("result merging" or "data fusion").

Resource selection requires information about the contents of each database (a "resource description"). In environments where databases are managed by independent organizations ("uncooperative environments"), the resource selection service must discover the contents of each database by *query-based sampling*, which is a process of submitting queries and observing what is returned; a surprisingly small number of queries suffices [3]. Documents sampled from each database are stored in a *centralized sample database*, and are used to generate database resource descriptions [3] and other information used to merge search results [7].

When databases contain very long documents, the costs of storing sampled documents can be high. For example, the U.S. Government Printing Office (USGPO) provides access to almost 2,400 databases of government publications. In tests with USGPO data the average length of documents obtained by query-based sampling was 440 kilobytes (KB). If 300 documents are sampled from each database (a typical sample size in our research), the size of a centralized sample database for this environment is 2,400 databases x 300 documents x 440 KB = 302 gigabytes. We would prefer it not to use so much space.

The research described here studied building resource descriptions and a centralized sample database from document summaries ("pruned documents"), instead of complete documents. The experimental results show that this approach allows a large decrease in the sizes of these data resources while reducing retrieval accuracy only slightly. The following section outlines different methods of summarizing documents. Experiments and results are presented in Section 3. Section 4 concludes.

2. Methods of Pruning Documents

We define *pruning* as removing words that are *not* selected from a document. The criteria used for pruning can be frequency-based or location-based. For *frequency-based* pruning term frequency

Rank	Full-content	Precision							
		TF	TFIDF	LUHNM	LUHNS	FIRSTM	FIRSTS	RANDM	RANDS
5	0.438	0.390 (-10.9%)	0.396 (-9.6%)	0.414 (-5.5%)	0.428 (-2.3%)	0.414 (-5.5%)	0.400 (-8.7%)	0.418 (-4.6%)	0.400 (-8.7%)
10	0.404	0.346 (-14.3%)	0.366 (-9.4%)	0.379 (-6.2%)	0.378 (-6.4%)	0.378 (-6.4%)	0.359 (-11.1%)	0.382 (-5.4%)	0.354 (-12.4%)
15	0.379	0.330 (-12.8%)	0.343 (-9.3%)	0.363 (-4.2%)	0.366 (-3.3%)	0.363 (-4.0%)	0.352 (-7.0%)	0.361 (-4.5%)	0.343 (-9.3%)
20	0.361	0.313 (-13.3%)	0.333 (-7.6%)	0.349 (-3.5%)	0.349 (-3.5%)	0.340 (-5.9%)	0.338 (-6.5%)	0.338 (-6.4%)	0.326 (-9.8%)
30	0.337	0.295 (-12.6%)	0.308 (-8.5%)	0.326 (-3.2%)	0.327 (-3.0%)	0.315 (-6.4%)	0.311 (-7.6%)	0.317 (-5.8%)	0.303 (-10.1%)
100	0.260	0.223 (-14.5%)	0.237 (-8.8%)	0.250 (-4.0%)	0.239 (-8.3%)	0.245 (-5.9%)	0.236 (-9.4%)	0.247 (-5.3%)	0.236 (-9.6%)

Table 3.1: Precision of document rankings created by the CORI result-merging algorithm on Testbed I.

Rank	Full-content	Precision							
		TF	TFIDF	LUHNM	LUHNS	FIRSTM	FIRSTS	RANDM	RANDS
5	0.436	0.388 (-11.0%)	0.370 (-15.1%)	0.442 (+1.4%)	0.404 (-7.3%)	0.442 (+1.4%)	0.374 (-14.2%)	0.430 (-1.4%)	0.382 (-12.4%)
10	0.411	0.371 (-9.7%)	0.352 (-14.3%)	0.426 (+3.6%)	0.388 (-5.6%)	0.426 (+3.6%)	0.369 (-10.2%)	0.411 (0.0%)	0.371 (-9.7%)
15	0.396	0.355 (-10.3%)	0.341 (-13.8%)	0.405 (+2.2%)	0.367 (-7.2%)	0.405 (+2.2%)	0.353 (-10.8%)	0.396 (0.0%)	0.355 (-10.4%)
20	0.388	0.353 (-9.2%)	0.341 (-12.2%)	0.401 (+3.1%)	0.366 (-5.7%)	0.400 (+2.9%)	0.352 (-9.5%)	0.391 (+0.7%)	0.353 (-9.1%)
30	0.364	0.327 (-10.1%)	0.316 (-13.2%)	0.371 (+1.8%)	0.339 (-6.9%)	0.369 (+1.4%)	0.328 (-10%)	0.361 (-0.9%)	0.331 (-9.1%)
100	0.274	0.249 (-9.0%)	0.246 (-10.0%)	0.277 (+1.2%)	0.251 (-8.3%)	0.277 (+1.2%)	0.248 (-9.5%)	0.270 (-1.3%)	0.249 (-8.9%)

Table 3.2: Precision of document rankings created by the CORI result-merging algorithm on Testbed II.

information is used to measure the importance of the terms and relatively unimportant terms or segments of the document are pruned. For *location-based* pruning terms are selected based on their positions in the document. Pruning methods can also be distinguished as *multiple-occurrence* or *single-occurrence* depending on whether multiple occurrences of a term are allowed in the pruned document.

Below we present eight methods of pruning documents. Four are frequency-based (TF, TFIDF, LUHNM, and LUHNS) and four are location-based (FIRSTM, FIRSTS, RANDM and RANDS). Three are multiple-occurrences methods (LUHNM, FIRSTM and RANDM) and five are single-occurrence methods (TF, TFIDF, LUHNS, FIRSTS and RANDS). The methods are discussed in detail in [6].

2.1 Pruning Based on Term Frequency (TF)

Given a threshold *r*, all unique non-stopword terms in the document are ranked by within-document term frequencies and terms ranked lower than *r* are discarded.

2.2 Pruning Based on Term Frequency and Inverse Document Frequency (TFIDF)

All unique non-stopword terms in the document are ranked by their tf.idf scores [6]. Given a ranking threshold *r*, terms ranked lower than *r* are discarded.

2.3 Pruning Based on Luhn's Keyword-Based Clustering (LUHNM and LUHNS)

Terms are ranked based on the importance of the sentence they occur. Luhn's keyword-based clustering is used to measure the importance of a sentence in the document [5, 6]. Each sentence has a score calculated based on the within-document term frequencies of the terms occurring in this sentence. Sentences are ranked by their scores. Given a threshold *t*, if multiple occurrences of a unique term are allowed, non-stopword terms are selected from top-ranked sentences until the number reaches *t* (LUHNM); otherwise, unique non-stopword terms from top-ranked sentences are selected until the number reaches *t* (LUHNS).

2.4 Pruning Non-First Part of the Document (FIRSTM and FIRSTS)

Given threshold *t*, the first *t* non-stopword terms are selected if multiple occurrences of a unique term are allowed (FIRSTM), or the first *t* unique non-stopword terms otherwise (FIRSTS). The rest are discarded.

2.5 Pruning Based on Randomly Selected Terms of the Document (RANDM and RANDS)

Positions in the document are chosen randomly and all corresponding non-stopword terms are selected if multiple occurrences of a unique term are allowed (RANDM), or those corresponding non-stopword terms that haven't been selected for this document otherwise (RANDS) until the number reaches *t*.

	Full-content	Multiple-occurrences methods	Single-occurrence
Threshold	N/A	400	100
Resource selection index size	24 MB	11 MB (54.17%)	9 MB (62.50%)
Centralized sample DB size	258 MB	46 MB (82.17%)	19 MB (92.64%)
Centralized sample index size	110 MB	43 MB (60.91%)	25 MB (77.27%)

Table 3.3: The sizes of data resources created from pruned and unpruned documents in Testbed I.

	Full-content	Multiple-occurrences methods	Single-occurrence methods
Threshold	N/A	1,600	400
Centralized sample DB size	51 MB	7 MB (86.27%)	2 MB (96.08%)

Table 3.4: The space taken by documents from two "very long documents" databases in the centralized sample database before and after content pruning in Testbed II.

3. Experiments and Results

3.1 Experimental Setting

The pruning methods were tested on two multi-database testbeds created from TREC data.

Testbed I consists of 100 databases obtained by dividing data from TREC CDs 1, 2, and 3 based on source and publication date. This testbed has been used in prior research [3, 7]. Title fields of TREC topics 51-150 and the standard relevance assessments supplied by NIST [4] were used.

Testbed II is a variation of Testbed I created specifically to simulate an environment like the USGPO where some databases have extremely long documents. Documents from the *1988 Federal Register* and the *1993 U.S. Patents* were concatenated to create very long documents based [6]. The resulting two databases had an average document length of 12,854 terms and 12,047 terms respectively. These two databases replaced 13 corresponding databases in Testbed I, resulting in Testbed II, which contains 89 databases. The relevance assessments for Testbed II were adjusted accordingly.

Resource descriptions and a centralized sample database were created using (pruned or unpruned) documents obtained by query-based sampling [3]. Databases were ranked with the CORI resource selection algorithm [3]. The 10 top-ranked databases for a query were selected for search. The selected databases were searched using the INQUERY search engine [1]. Document rankings produced by different databases were merged into a single ranking by the CORI result-merging algorithm [3] or a regression-based Semi-Supervised Learning (SSL) result-merging algorithm [7].

For Testbed II, sampled documents were pruned only if they belong to the two "very long documents" databases to guarantee that any differences in performance would be due only to pruning very long documents. The pruning thresholds (the number of terms to be selected for a document) were 1,600 for multiple-occurrences and 400 for single-occurrence methods. For Testbed I, all sampled documents were subject to pruning. The pruning thresholds were 400 for multiple-occurrences methods and 100 otherwise.

The retrieval performance was measured by precision at document ranks 5, 10, 15, 20, 30, and 100.

3.2 Effects of Document Pruning on Overall Retrieval Performance

A series of experiments was conducted to study the effects of the pruning methods. Tables 3.1 and 3.2 summarize the experimental results using the CORI result-merging algorithm. Results using the Semi-Supervised Learning result-merging algorithm are similar and are omitted here due to space limits. The baseline results were generated from unpruned documents.

The LUHNM, LUHNS, RANDM, and FIRSTM pruning methods caused less than a 10% loss in relative accuracy; usually the difference was less than 5%, which few users would notice. The loss of Precision

using other methods was usually less than 15%. Although multiple-occurrences methods generally provided better accuracy (less degradation) than single-occurrence methods, the thresholds for single-occurrence methods were more aggressive, and produced smaller storage costs.

These experimental results suggest that resource descriptions and a centralized sample database created from pruned documents are as effective as those constructed from complete documents.

3.3 Effects of Document Pruning on Storage Costs

Three types of data were affected by pruning: i) the resource selection index, ii) the centralized sample database, and iii) the centralized sample index. Table 3.3 compares the average sizes of these data resources before and after pruning on Testbed I. The percentage figures indicate the reduction in disk space relative to the (unpruned) baseline. Table 3.4 compares the space required to store the sampled documents from two "very long documents" databases in Testbed II before and after pruning.

These results show dramatic reductions in the storage costs of three different types of data used in federated search by pruning sampled documents.

4. Conclusions

In a federated search system sampled documents were used to build resource descriptions for resource selection, and a centralized sample database for result merging. The research reported here studied several approaches to pruning documents obtained by query-based sampling to reduce disk storage costs. Our results demonstrate that pruned documents can reduce storage costs 54-93%, which is a considerable savings when documents are very long, with only minor degradation in search accuracy. Location-based pruning methods worked well in our tests, even though they ignore frequency. In general, LUHNM, LUHNS, FIRSTM and RANDM pruning give good and stable performance.

Acknowledgements

This material is based on work supported by NSF grant EIA-9983253. Any opinions, findings, conclusions or recommendations expressed in this material are the authors', and do not necessarily reflect those of the sponsor.

References

[1] J. Allan, J. Callan, M. Sanderson, J. Xu and S. Wegmann. INQUERY and TREC-7. In *Proc. of the Seventh Text Retrieval Conference (TREC-7)*.

[2] J. Callan, Z. Lu and W. B. Croft. Searching distributed collections with inference networks. In *Proc. of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1995.

[3] J. Callan. Distributed information retrieval. W. B. Croft, editor, *Advances in information retrieval*, chapter 5. Kluwer Academic Publishers, 2000.

[4] D. Harman, editor. *Proc. of the Third Text Retrieval Conference (TREC-3)*. 1995.

[5] A. Lam-Adesina and G. Jones. Applying summarization techniques for term selection in relevance feedback. In *Proc. of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2001.

[6] J. Lu and J. Callan. Pruning long documents for distributed information retrieval. In *Proc. of the Eleventh International Conference on Information Knowledge Management* (CIKM 2002). 2002.

[7] S. Luo and J. Callan. Using sampled data and regression to merge search engine results. In *Proc. of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2002.