Results Merging in a Distributed Search System

V.V. KLUEV
The Core and Information Technology Center
The University of Aizu
Tsuruga Ikki-machi Aizu-Wakamatsu City
Fukushima 965-8580
JAPAN

Abstract: Mechanisms of results merging in a distributed search system have been discussed in this paper. Such mechanisms' aims are to select the most relevant documents retrieved by different servers and put them on the top of the list returned to the end u ser. In our calculation, we use the clustering technique to divide retrieved results into several homogeneous groups and a metric on the base of the vector space model to arrange items inside each group. The main thrust of our approach is both the proposed clustering metric and the ranking metric. Our mechanisms have been implemented in the OASIS system, a system which belongs to the distributed search systems in the Internet.

Key-Words: - Results Merging, Search Engine, Distributed System, Metasearch, Ranking

1 Introduction

It is well known that the Internet continues to expand rapidly. There are dozens of powerful search tools on the net available without cost. Nevertheless, searching for information is still inefficient. How is it possible to find app ropriate information easily and quickly? This main question is still unanswered. There are a plenty of explanations on the net how to search. One of them can be found at [3]. The current trend in developing new tools is to get more and more instruments designed on the base of the distributed architecture. The main advantages of a distributed system against a system using the centralized architecture are as follows:

- Administration of a system should be easier.
- The index of a whole system should be larger when compared to other approaches.
- Results of a search have to be more accurate.

The key problems of distributed searches have been discussed in the following citations [2, 4 5, 8, 10, 12, 13, 15]. Problems pointed to are reflected here:

1) How can a search a ccurately determine a small number of potentially useful collections (local search engine) to invoke for each user query?
2) How can an engine search inside this kind of collections consisting of topic identical documents for relevant ones?

The answers to the se questions depend on the nature of the system. Distributed systems can be classified as follows:

1) General systems. The space has been divided on the base geographical principle. For example, one server can index documents from sources located in central Europe; the responsibility of the second one is Japan, etc. The classical example is the HARVEST system [10].
2) Meta systems. These form a separate type in the distributed class systems. They usually support unified access to multiple local search engines. T hese systems do not maintain their own index on web pages. Particular systems can be used as extensions to the most popular powerful search engines. For example, SearchPad is a c lient side meta search tool that searches all major online search engines at o nce using definitions preset by the user that allow for incredibly exact matches [2].
3) Topic specific systems. Each local server has collected documents relative to a special topic. For example, the topic of search may be algorithms, sightseeing, etc. OASIS is an example of this kind of system [1].

A common and difficult task for each kind of systems is results merging. The aim of this process is to combine results returned from selected

collections (local search engines) into a final single ranked list. Th e major obstacles toward resolving this task are as follows:

- The different local servers can return the same documents because the servers' databases intersect. Duplicates have to be eliminated.
- Metrics being used by different servers cannot be correlated, even servers use the same ranking method. The full set of returned documents has to be evaluated somehow before presenting the results to the end user.

In this paper, we present a new technique to resolve the aforementioned tasks. This technique has shown promising results. It has been applied to the OASIS system [1].
The rest of the paper has been organized as follows. Section 2 describes some well -known techniques for results merging. Section 3 presents our approach in detail. Section 4 discusses the details of evaluations we carried out in our testbeds. The final section gives comments on experimental results.

2 Related Work

The task of merging results was discussed in several citations [2, 4, 5, 6, 7, 8]. Proposals with different solutions can be fo und there. In this section, we will review them shortly.
In respond to a query each server (collection) returns a list of documents retrieved. This list usually consists of the following items:

- URL;
- The title of the document;
- A short description of the doc ument (the aim of this item is to help the user to estimate the document);
- A document score calculated according the measure applied on the server;
- Auxiliary information: specification of the document format, date of indexing, document size, etc.

2.1 Duplicate Recognition

The simplest variant of the duplicate recognition task is detection of the same URLs. This is not easily accomplished if documents are located on different servers ("mirrors") and presented in different formats (for example ps, pdf, html or text). There is not adequate information to make a decision. In any case, a very simple method proposed in [9] can be applied. If names of document files differ only in suffixes documents can be considered as the same.

2.2 Ranking Strategy

Current appr oaches to combine results returned by different search engines into a single ranked list can be distinguished in the following way:

1) Row score merging;
2) Round robin merging;
3) Merging on the base of collection scores;
4) Merging on the base of global similarities;
5) Clustering.

Authors of the first approach [6] assume that document collections have been indexed using the same model and document scores obtained as a result of the search are comparable across all collections. These scores are then used to merge search results into a single list.
The basic assumption for the second approach looks like this: Each document collection consists of approximately the same number of relevant items and they are equally distributed within results lists [4]. From this hypothesis a final results list can be created in a round robin manner.

The systems using the third merging strategy compute a score for each collection and for a given query. This collection score is then utilized to modify the score attached to each document. The detailed description and necessary formulas can be found in [10].

As noted in [5], three aforementioned approaches are all heuristics and they cannot guarantee the selection of all potentially useful documents for a given query.

An approach based on a g lobal similarity threshold aims to retrieve all potentially useful documents from each selected collection. The problem is that documents (full content) may need to be fetched to the search system to enable the computation of their global similarities. Th e advantage is that high quality merging can be achieved [5].

The clustering strategy has been applied to the OASIS system. Its aim is to split results into various sets (clusters). The documents inside each group are similar to each other by topics they are related to. One representative from each cluster has been included into a final results list. The clustering is executed by the competitive learning neural network [1]. The disadvantage of this solution is that huge document profiles are sent though the

Internet. Moreover, LSI (Latent Semantic Indexing) similar calculation is running before clustering to reduce the size of document profiles. As a result, it decreases the performance of the system. Collected statistics show: A neural network does not pro duce well split clusters.
The main problem in information retrieval is the lack of a powerful language model. From this, most of application methods are empirical and based on heuristics. As a result, in our case, it is practically impossible to compare th e aforementioned approaches because they were designed for different systems and were tested using different datasets.

3 Mechanisms Used

3.1 Common Note

The basic assumptions for our approach are as follows. Users usually interact with a search system several times to find information they look for. For each chunk of information need they may submit several queries, often by a process of query refinement. The aim of each iteration is to narrow the search topic and discard unnecessary documents. The best strategy for a system is to split returned results into the various sets and deliver to the user the most important representatives from these sets without losing relevant data. The system should minimize the volume information sent though the Internet d uring every iteration. The key ideas from the approaches mentioned in Section 2 have been utilized here. Our proposed mechanisms take into account:

- The title of the every returned document;
- A selected part of the document text to browse by a user;
- Size of the document in bytes;
- Score for each selected collection for query propagation;
- Term frequency of each query term in every returned document;

The first three components are a standard de facto for browsing in the search engine world. The usage of the la st two will not significantly increase the volume of information being sent though the Internet.

3.2 Clustering Scheme

It is known that clustering is a common way to divide retrieved documents into several sets. A number of studies [13, 14] have reporte d on experiments related to an automatic text clustering technique. Researchers [12] tested Ward's clustering and obtained promising results. We adopted a technique used by them. The main clustering algorithm is very simple, and it is practically identical to Ward's technique.

Description of the Algorithm: Let X be the set of documents to be clustered; let C be the set of clusters and N be the predefined number of clusters (the number of returned results to be present to the user). The algorithm starts wi th a separate cluster for each document from X. In each step, the two most similar clusters C_i and C_j are determined and merged into a new cluster C_{new}. The algorithm terminates when a predefined number of clusters N containing all documents from X has been formed.

Metrics and thresholds used in these calculations are key components. Our metrics is based on the widely used vector space metric. In pursuance of [11] we estimate a number of words in the document written in English as:

$$WN = \frac{CN}{5}$$

where CN is the size of the document in bytes.
All aforementioned components from document description returned by the collection have been used in our distance metric. This description was considered as a document. We modified the standard $tf * idf$ formulas to compute term weights [16]: the final term weight is the product of $tf * idf$ weight and the collection score from the retrieval source of the document:

$$w\_final_{ij} = w_{ij} * col\_score_j$$

where

- w_{ij} is $tf * idf$ weight of term i in document j
- col_score_j is a collection score (a collection selection subsystem calculates this score). Note that the collection score is less that 1.

3.3 Ranking

To make a final decision about document relev ance scores, we modified the formula proposed in [8].

$$score(D,Q) = \frac{nb_q + c_1 * nb\_occ + c_2 * doc\_score + c_3 * col\_score}{\sum_i score(D_i,Q)} (*)$$

We agree with authors [8] that the number of query terms included in each document and frequencies of query terms in documents are good relevant indicators. The following components should be taken in to account as well: a) the collection scores and b) the document scores assigned by the collection from which the document was retrieved. On the basis of this assumption, we calculate the document score according formula (*). Parameters for each document D are as follows:

- nb_q is the number of query terms presented in D;
- nb_occ is the total number of occurrences of query terms in D;
- doc_score is a document score assigned by the collection from which the document was retrieved;
- col_score is a collection score (a collection selection subsystem calculates this score);
- c_1, c_2 and c_3 are constants, which are set to 0.100.

Aforementioned formula is applied to each cluster centroid. Documents presented to the user are arranged according to their scores.

4 Results of the Tests

The proposed approach has been tested using the OASIS system. These tests are discussed in this section. The following configuration of the system was dedicated for our tests: Three local servers (the installation place being Aizu Wakamatsu City) and one remote server (at Koriyama City). The distance between these cities is about 60 km. The test topic specific collections [9, 17] consisting of the real Internet data were involved in our experiments. Table 1 describes a distribution of document collections installed on servers. The aim of these tests is to compare the previously implemented methods with proposed ones in this paper.

We assume that the following are a good accuracy indicator of the merging technique: the number of actually relevant documents on the top of the list presented to the user, and the number of actually relevant documents in each compiled cluster as well. These parameters are very important from the user's point of view. We used a set of short queries, similar to those submitted to the search engine. They consist of one, two and three words. These queries reflect the real search process on the Web. We conducted our tests as follows:

1) A query was submitted to the system;
2) The system fetched results of the search;
3) After deleting the duplicates returned results were clustered;
4) Items inside each cluster were arranged according to relevance score;
5) Centroids of 10 clusters were submitted to the user;
6) The user could see up to three documents from every cluster;
7) The evaluation of the results accuracy was done only by human inspection.

Table 1 Location of the collections

Servers	Collections	Number of documents
Aizu: 1	Programming Languages	7659
Aizu: 2	Algorithms	7775
Aizu: 3	Travelogues	226
	Linux & Unix	488
	Information Retrieval	202
	Research Groups	811
	Physics	467
	Card Games	798
	Museum	444
	Monitors	70
Koriyama	Programming Languages	445
	Cars	427

Table 2 shows the main results of our tests.

The system retrieved results for the same query set using the older merging method and the system presently being discussed. In our tests, document sets before a merging process were the same for both methods. The number of retrieved documents by each server was between 0 and 20. As we can see, columns 2 – 4 present the length of queries in words; columns 5 and 7 show the number of relevant items among the top three documents from the list presented to the user. Columns 6 and 8 point to a quote of relevant documents among the top five. In general, the new method produces better results. The amount of information sent via the net was reduced significantly without losing the

Table 2. Results of the search

Topic	Query length			Search results			
	1 W	2 W	3 W	Old method		New method	
				Top 3	Top 5	Top 3	Top 5
1	2	3	4	5	6	7	8
Programming Languages	5	1	4	2	3	3	3
Algorithms	4	2	1	2	2	2	3
Cars	1		2	1	2	2	3
Travelogues			2	0	1	1	1
Linux & Unix		2		0	1	0	1
Information Retrieval		2	1	0	1	0	2
Research Groups	1			0	0	0	1
Physics		2		1	0	0	1
Card Games	2	1	2	2	2	1	2
Museums	4	3	2	3	3	3	3
Monitors		1		0	1	0	0

accuracy of the search. The second advantage is in reducing calculations to merge the results. Our new clustering algori thm runs in $O(N^2)$ time [15]. Using this algorithm we discarded the time consuming clustering by neural networks [1] and time consuming LSI calculation as well. The new method showed the poorer results than the older one in the case of sea rching for information related topics of Monitors, Card Games and Physics. Both methods produced the same results for topics of Museums and Linux&Unix. The documents that are the closest to the centroids were also analyzed. The number of relevant hits from them is the same on average for both methods (2 – 3 using each methods).

5 Conclusion

This paper introduces a new technique to merge search results in distributed systems. It utilizes the idea of preliminary clustering and then arranging items inside ea ch cluster. Cluster representatives can be submitted to the user in the usual search engine manner. All calculations are based on standard de facto information: the document title, its size, a selected part of the text to browse by a user, etc. From these calculations, this technique can be applied to metasearch systems as well. This technique was incorporated into the OASIS system. Results shown during our preliminary tests are promising. They are qualitatively at least as good as results produced by the time consuming neural network clustering and LSI calculations.

References:

[1] *OASIS: Distributed Search System in the Internet,* Edited by A. Patel, L. Petrosjan, W. Rosenstiel, St. Petersburg: St. Petersburg State University Published Press, 1999, - 614 p. (ISBN 5-7997-0138-0)

[2] Krishna Bharat, SearchPad: Explicit Capture of Search Context to Support Web Search, *In Proceedings of 9th International World Wide Web Conference* , Amsterdam, May 15 – 19, 2000.

[3] http://searchwebmanagement.techtarget.com/sDefinition/0,,sid27_gci212955,00.html

[4] Voorhees E.M., Gupta N.K., and Jonson - Laird B., Learning Collection Fusion Strategies, *in Proceedings of the ACM SIGIR'95*, 1995, pp. 172 – 179.

[5] Zonghuan Wu, Weiyi Meng, Clement Yu, Zhuogang Li, Towards a Highly -Scalable and Effective Metasearch Engine, *In Proceedings of 10th International World Wide Web Conference*, 2001

[6] Kwork K.L., Grunfeld L., and Lewis D.D., TREC-3 Ad -hoc: Routing R etrieval and Thresholding Experiments Using PIRCS, *In Proceedings of TREC-3, 1995, pp. 247- 255.*

[7] L. Gravano, C. Chang, H. Garcia -Molina, and A. Paepcke. *STARTS: Stanford Proposal for Internet Meta -Searching*. ACM SIGMOD Conference, Tucson, May 1997, pp. 207-218.

[8] Yves Rasolofo, Faiza Abbaci, Jacques Savoy: Approaches to Collection Selection and Results Merging for Distributed Information Retrieval, *CIKM 2001* , pp. 191 - 198

[9] V. Kluev, Compiling Document Collection from the Internet, *ACM SIGIR Forum* , Vol. 34, Number 2, Fall 2000, pp. 9 – 14.
[10] Callan J. Distributed Information Retrieval. *In Advances in Information Retrieval, Edited by W.B. Croft* , Kluwer Academic Publisher, 2000, pp. 127 – 150.
[11] ReadRace http:// www.epixmed.com/hunsocma/ReadRace/
[12] Roussinov D.G., Chen H, Document Clustering for Electronic Meetings: An Experimental Comparison of Two Techniques, *Decision Support Systems* , 27 (1999), pp. 67 - 79.
[13] Martin Ester, Hans -Peter Kriegel, Jourg Sander, Miha el Wimmer, and Xiaowei Xu, Incremental Clustering for Mining in a Data Warehousing Environment, *in Proceedings of the 24th VLDB Conference*, USA, 1998.
[14] Christopher D. Manning and Hinrich Schutze, *Foundations of Statistical Natural Language Processing* , The M TI Press, England, 2000, 680 p.
[15] Larkey L.S., Connel M.E., and Callan J, Collection Selection and Results Merging with Topically Organized U.S. Patents and TREC Data, in Proceedings of CIKM'200, pp. 282 – 289.
[16] G.Salton, *Automatic text processing: the transformation, analysis, and retrieval of Information by computer* . Reading, MA:Addison-Wesley, 1989.
[17] V.Kluev, Source Selection in a Distributed Search System, *in V.Kluev, N.Mastorakis, editors, Topics in Applied and Theoretical Mathematics and Computer Science* , WSEAS Press, 2001, pp. 293 – 298.