Discovering the Representative of a Search Engine

King-Lup Liu,
School of CTI
DePaul University
Chicago, IL 60604
kliu@cs.depaul.edu

Clement Yu
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607
yu@eecs.uic.edu

Weiyi Meng
Department of Computer Science
SUNY-Binghamton
Binghamton, NY 13902
meng@cs.binghamton.edu

ABSTRACT

Given a large number of search engines on the Internet, it is difficult for a person to determine which search engines could serve his/her information needs. A common solution is to construct a metasearch engine on top of the search engines. Upon receiving a user query, the metasearch engine sends it to those underlying search engines which are likely to return the desired documents for the query. The selection algorithm used by a metasearch engine to determine whether a search engine should be sent the query typically makes the decision based on the search-engine *representative*, which contains characteristic information about the database of a search engine. However, an underlying search engine may not be willing to provide the needed information to the metasearch engine. This paper shows that the needed information can be estimated from an uncooperative search engine with good accuracy. Two pieces of information which permit accurate search engine selection are the number of documents indexed by the search engine and the maximum weight of each term. In this paper, we present techniques for the estimation of these two pieces of information.

Categories and Subject Descriptors

H.3 [**Information Systems**]: Information storage and Retrieval

Keywords

Search engine, metasearch engine, database size, term weight

1. INTRODUCTION

The Internet has become a vast information resource in recent years. To help ordinary users find desired data in this environment, many *search engines* have been created. Each search engine has a *text database* that is defined by the set of documents that can be searched by the search engine. Frequently, the information needed by a user is stored in the databases of multiple search engines. To facilitate a user to find the desired information, a common solution is to implement a *metasearch engine* on top of many local search engines. A metasearch engine is essentially an interface. When it receives a user query, it first passes the query to the appropriate local search engines, and then collects (sometimes, reorganizes) the results from its local search engines. With such a metasearch engine, only one query is needed from the user to invoke multiple search engines.

To avoid wasting resources, for a given query, a sophisticated metasearch engine invokes only those search engines that are most likely to provide the desired documents. Typically, a metasearch engine identifies such search engines based on some characteristic information of each underlying search engine. We call such characteristic information of a search engine the *representative*. The information kept in a representative depends on the approach used by the metasearch engine for selecting useful search engines[5]. To find an effective and efficient method to select text databases to search has been one of our research goals. We provided several solutions to this problem [2, 3, 4, 9, 10, 11]. Our experimental results showed that our solutions achieved near optimal results. As with most approaches to the database selection problem, we implicitly assume that the underlying local search engines are cooperative and are willing to provide the information needed by the metasearch engine. Under this assumption, for each local search engine, the representative acquired by the metasearch engine would faithfully reflect its contents. However, in the Internet environment, each search engine is usually autonomous and managed with its own interest in mind. The contents of each search engine may be viewed as proprietary. Thus, a local search engine may not be willing to provide all the information requested by the metasearch engine. It may even provide information that leads to an incorrect/inaccurate representation of its contents. In this paper, we show how the information needed to construct a search-engine representative may be obtained or estimated. Our method uses a sampling approach. Documents are sampled from a local search engine. No special cooperation is needed from the local search engine. Note that the information contents of a search-engine representative depends to a large extent on the approach to selecting useful search engines and can be very detailed. In this paper, we estimate the following quantities:

- *the number of documents indexed by a search engine*
  It was pointed out in [1] that this estimation is an open problem.

CIKM'02, November 4–9, 2002, McLean, Virginia, USA.

- *the maximum weight of each term in the vocabulary of a search engine*
  In our earlier work [2, 4, 9, 10, 11], we showed that the use of the maximum weights of terms permitted optimal retrieval results for single-term queries and near optimal results for multiple-term queries.

The rest of this paper is organized as follows. In section 2, we discuss our sampling technique for estimating the number of documents indexed by a search engine. We give the experimental results for a text database. Section 3 describes how we estimate the maximum value of the global weights of a term in all the documents of a local search engine and explains the rationale behind our technique. We summarize and conclude in section 4.

2. ESTIMATING THE NUMBER OF DOCUMENTS INDEXED BY A SEARCH ENGINE

Our search-engine selection method, as well as various other search-engine (database) selection methods, needs to know the number of documents indexed by each underlying search engine. Whether the number of documents indexed by a search engine can be estimated by sampling its documents is an open problem for some time [1]. In this section, we show that the number of documents indexed by a search engine can be estimated with good accuracy by sampling.

Let N be the number of all documents indexed by a search engine. We first draw a random sample of m documents from the search engine. Suppose we choose a document randomly from the collection of all documents indexed by the search engine. Then, the probability that this document is from the sample of m documents is $\frac{m}{N}$. Now, we perform n times the process of randomly selecting a document and observing whether it is a document from the earlier sample. Suppose Y of these n randomly selected documents are from the sample of m documents chosen earlier. From probability theory, *E(Y)*, the expected value of Y, is $n \times \frac{m}{N}$. By taking the observed value Y_o of Y in the sample as an approximation to the expected value *E(Y)*, i.e., $Y_o \approx n \times \frac{m}{N}$, we then obtain an estimated value $n \times \frac{m}{Y_o}$ of N, the number of documents indexed by the search engine.

To test our techniques, we formed three text databases of different sizes and applied our techniques to estimate their sizes. The documents are from the TREC collection. The first database has 100,000 documents, the second 200,000 and the third 300,000. For each text database, we drew several samples of varying sizes. For each sample, we performed the experiment a number of times. Each time a different number of documents was examined. The estimation results of the size of the third database is shown in table 1. The last row of the table gives the number of documents randomly picked for examination (i.e., the value of n). The entries of the first column, except the last entry, specify the sample sizes. The entries of the other columns, except those in the first and last rows, are the percentage errors of our estimated values of the database size.

Judging from our experimental results, our technique works reasonably well. For all three text databases, we obtained a percentage error of less than 2.5% of the database size by checking a total of no more than 2.5% of all the documents.

Sample size m	% error	% error	% error
1000	4.8 %	4.3%	3.7%
2000	4.4%	3.7%	3.4%
3000	3.9%	3.3%	2.9%
4000	3.4%	3.0%	2.5%
No. of docs examined n	1000	2000	3000

Database size = 300,000

Table 1: Estimation Results for a text database

3. ESTIMATION OF MAXIMUM GLOBAL TERM WEIGHT IN A LOCAL SEARCH ENGINE

The weight of a term in a document is a measure of the significance of the term in representing the document. Each search engine determines the weight of a term using its own term-weighting formula. The weight of a term in a document computed using the term-weighting formula of a local search engine shall be referred to as *local weight* of the term and that determined by the term-weighting formula of the metasearch engine as the *global weight* of the term.

When a query is submitted to a metasearch engine, in determining whether a local search engine should be searched or not, we have shown that the maximum values of the global weights of the query terms in all the documents indexed by the local search engine are critical information [2, 4, 9]. We refer to the maximum value of the global weights of a term in all the documents indexed by a local search engine as the *maximum global weight* of the term in the local search engine. Note that a local search engine is usually autonomous. It may not be willing to expend extra resources to compute the maximum global weight of each term in the local search engine using the term-weighting formula of the metasearch engine. In this secton, we discuss how the maximum global weight of a term in a local search engine may be estimated.

The problem of finding the maximum value of a dataset is trivial if the values of the dataset are known. A single pass of the values is sufficient to find accurately the maximum value of the dataset. However, the global weights of a term t in the documents indexed by a search engine are not known. To determine accurately the maximum global weight of a term t in a search engine, we need to download all the documents containing the term t and compute the global weight of term t for each downloaded document. As there are many such documents and there are many terms in the vocabulary of the search engine, repeating this process for each term is computationally not feasible. A method was developed for finding an accurate estimate of the maximum global weight of each term t. It is based on the observation that for many combinations of global term-weighting formula and local term-weighting formula, documents that have relatively large local weights for a term tend to have relatively large global weights for the term.

We formed 20 text databases for 20 hypothetical local search engines. The documents are from the TREC collection. The total number of documents in these text databases is 550,000. That is, the hypothetical metasearch engine has 550,000 documents. The global document frequencies of terms are determined from all the documents in the 20 text databases. For our experiments, we chose 5 local text databases, each having 27,500 documents. Identical experiments were performed on these databases. We then averaged the experimental results obtained.

For each combination of local term-weighting formula and global term-weighting formula, we perform the following steps:

1. chose randomly 200 terms in the local database;
2. for each term t chosen in step 1,
   (a) computed the local weight and global weight of term t in each document in the local database having term t; and determined the actual maximum global weight of term t in the local database;
   (b) obtained 30 documents with highest local weights of the term;
   (c) determined the maximum of the global weights of term t in (i) the 20 documents with highest local term weights and (ii) the 30 documents with highest local term weights;
   (d) for the maximum value of the global term weight obtained in each of the two cases (i) and (ii) of the previous step, computed the ratio of the maximum value to the actual maximum global weight of the term in the local database (obtained in step 2.(b)); and if the computed ratio is at least 0.99, we recorded that a *sufficiently accurate estimate* has been obtained.

We used two well-known classes of term-weighting formulas: the Okapi term-weighting formula[6] and the *tf-idf* term-weighting formula[8, 7]. We performed extensive experiments using different combinations of different variations of both classes of formulas. Our experimental results show that on average, the number of terms (out of 200) for which sufficiently accurate estimates have been obtained for their maximum global weights was (a) 187.32 when the top 20 documents were sampled, and (b) 190.66 when the top 30 documents were sampled.

4. CONCLUSIONS

Deciding on which local search engines to search for a given query is an important component of a metasearch engine, especially when the metasearch engine has a large number of underlying search engines. Typically, the decision is made based on the search engine representative, which contains characteristic information about the database of the search engine. Two pieces of information which permit accurate search engine selection are the number of documents indexed by the search engine and the maximum weight of each term. In this paper, we presented techniques for the estimation of these two pieces of information.

In [1], it was pointed out that the estimation of the number of documents indexed by a search engine is an open problem. We developed a technique that makes this estimation possible. The number of documents indexed by a search engine is estimated by drawing a random sample of its documents. Three text databases were formed using the text documents from the TREC collection. Our technique was applied to estimate their sizes. In each case, an estimation accuracy of less than 2.5% estimation error of the database size was achieved by sampling no more than 2.5% of the documents in the database.

The *global* weight of a term in a document is the weight of a term in a document determined by a metasearch engine and is usually different from the weight of the term in the same document determined by a local search engine. We are interested in estimating the maximum value of the global weights of a term in all documents indexed by a search engine. Our approach is to sample the top 20 or 30 documents with the highest local term weights. We performed experiments using the Okapi and *tf-idf* term-weighting formulas. Our experimental results showed that for more than 90% of the terms tested, the estimated maximum global weight deviated from the actual maximum global weight by less than 1%.

5. ADDITIONAL AUTHORS

6. REFERENCES

[1] J. Callan, M. Connell, and A. Du. Automatic discovery of language models for text databases. In *Proceedings of ACM SIGMOD*, pages 479–490, 1999.

[2] K. Liu, C. Yu, W. Meng, W. Wu, and N. Rishe. A statistical method for estimating the usefulness of text databases. *IEEE Transactions on Knowledge and Data Engineering.* (to appear).

[3] W. Meng, K. Liu, C. Yu, X. Wang, Y. Chang, and N. Rishe. Determining text databases to search in the internet. In *VLDB*, 1998.

[4] W. Meng, K. Liu, C. Yu, W. Wu, and N. Rishe. Estimating the usefulness of search engines. In *ICDE*, March 1999.

[5] W. Meng, C. Yu, and K. Liu. Building efficient and effective metasearch engines. *ACM Computing Surveys*, 34(1):48–89, March 2002.

[6] S. Robertson, S. Walker, and M. Beaulieu. Okapi at trec-7: automatic ad hoc, filtering, vlc and interactive. In *Overview of the Seventh Text Retrieval Conference*, 1998.

[7] G. Salton and M. McGill. *Introduction to Modern Information Retrieval.* McCraw-Hill, New York, 1983.

[8] S. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[9] C. Yu, K. Liu, W. Wu, W. Meng, and N. Rishe. Finding the most similar documents across multiple text databases. In *Proceedings of the IEEE Conference on Advances in Digital Libraries (ADL'99), Baltimore, Maryland*, May 1999.

[10] C. Yu, W. Meng, K. Liu, W. Wu, and N. Rishe. Efficient and effective metasearch for a large number of text databases. In *Proceedings of ACM CIKM*, November 1999.

[11] C. Yu, W. Meng, W. Wu, and K. Liu. Efficient and effective metasearch for text databases incorporating linkages among documents. In *Proceedings of ACM SIGMOD*, pages 187–198, 2001.