Discovery of Similarity Computations on the Internet

K.L. Liu, C. Yu
Department of EECS,
University of Illinois at Chicago
Chicago, IL 60607
(312)996-2318
kliu, yu@eecs.uic.edu

Weiyi Meng
Department of Computer Science
SUNY -- Binghamton
Binghamton, NY 13902
(607)777-4311
meng@cs.binghamton.edu

N. Rishe
School of Computer Science
Florida International University
Miami, FL 33199
(305)348-2025
rishen@cs.fiu.edu

1. MOTIVATION

The World Wide Web can be viewed as a huge digital library. A search engine is a typical means to tap into this vast source of information. There are several situations in which it is of interest to discover the formula used by a search engine to determine the similarity of a document to a query. One such situation is as follows. Currently, there are numerous search engines available on the Internet and the number is likely to increase many times in the future. Finding the most desired documents can be a formidable task as these documents may be located in different search engines. To facilitate the retrieval of documents, a solution is to construct a global search engine, also known as metabroker or metasearch engine, on top of the (local) search engines [3, 7, 9]. One of the challenging problems of building an efficient and effective global search engine is the *document selection problem* (discussed below). Our solution to this problem requires that the global search engine know how a local search engine assigns similarities to documents with respect to a given query. When such information is not available, the global search engine may need to employ some means to discover how a local search engine determines document similarities.

A global search engine is essentially an interface, which logically contains all documents in the underlying local search engines. However, the actual documents are not stored in the global search engine. A user submits his/her query to and obtains the retrieved documents from the global search engine. Upon receiving a query Q, the global search engine dispatches Q to the local search engines and requests that the desired documents be retrieved. To avoid wasting resources, usually the global search engine sends Q only to those local search engines most likely to contain the desired documents. Then, the query results from the chosen local search engines are collected and the n most similar documents to query Q are returned to the user.

When a search engine receives a query Q, it computes for each document d a similarity value $sim(Q, d)$ between Q and d using some similarity function. A document d is judged to be desired or similar if the similarity value $sim(Q, d)$ is high or, more specifically, is above a given threshold. Note that the local search engines underlying a global search engine are usually autonomous. With respect to a given query, the similarity of a document computed by a local search engine is likely to be different from that of the same document determined by the global search engine. Documents considered to be most similar to Q by the global search engine may have low similarities in the local search engine and hence are not retrieved. The problem of retrieving (globally) desired documents from local search engines is the *document selection problem* and has received much interest recently [1, 2, 4, 6, 8]. In [6], for a given query Q and a given global similarity threshold, we presented a technique to determine an optimal local similarity threshold for each chosen local search engine L. This optimal local threshold for L guarantees that all documents which are globally most similar to query Q will be retrieved while minimizing the retrieval of non-similar documents from L. This technique of setting the optimal local threshold requires that the global search engine know how the similarity of a document to a query is assigned in the local search engine [6].

2. THE PROBLEMS

Conceptually, each document d is represented as a vector of weights $(w_1, \ldots, w_m)$, where w_j is the weight of term t_j in representing d. A query Q is also represented as a vector of weights $(u_1, \ldots, u_m)$. The dot-product similarity function is a commonly used similarity function. (The well-known Cosine similarity function is simply the dot-product function with the weights of terms computed in a specific manner.) It determines the similarity $sim(Q, d)$ between Q and d as $(u_1w_1 + \ldots + u_mw_m)$. Assuming that a search engine uses the commonly used dot-product similarity function, the formula employed by a search engine to compute document similarities can be determined if we know how it assigns weights to terms in documents and a query. However, in the Internet environment, the weight of each term used by a retrieval function of a search engine can be unknown as the information is proprietary. We are interested in developing techniques to find out how the term weights are assigned by a search engine.

There are several components affecting the weights of terms (e.g., term frequency, document frequency and norm). The mathematical expression for each component frequently involves a number of constants that can be adjusted to improve the retrieval effectiveness of a search engine. Our task is to find out for each term-weight component

(a) the form of the mathematical expression used by a search engine, and

(b) the values of any constants embedded in the mathematical expression.

DL 99, Berkeley, CA USA

3. OUR TECHNIQUES

Our methodology for discovering for each term-weight component the mathematical expression employed by a search engine consists of the following steps.

(a) Create a knowledge base of different known mathematical expressions. We achieve this by surveying research papers and reports.

(b) Form a set of carefully designed probe queries and submit them to the local search engine.

(c) Analyze the retrieval results to determine which mathematical expression in the knowledge base is used. If no expression in the knowledge base is found to be the correct mathematical expression, then we attempt to create a new expression that can explain the retrieval results. If such an expression is found, it is added to the knowledge base.

(d) Determine the values of any constants embedded in the identified mathematical expression.

After a user submitted a query to a search engine, the retrieved documents are usually ranked in non-increasing order of their similarity values to the query before they are presented to the user. However, the similarity of each retrieved document may or may not be provided to the user. The techniques we developed only make use of the rank order of the retrieved documents and do not require that the document similarities be known. A systematic way to solving the constants embedded in the mathematical expressions of the term-weight components is given in [5].

4. EXPERIMENTAL RESULTS

We have experimented on the WebCrawler news search engine using our current techniques. We estimated the formulas employed by WebCrawler for assigning weights to terms. To test how accurate the estimated term-weight formulas approximate the actual formulas used by WebCrawler, we submitted a set of 50 queries to its news search engine. For each of these queries, more than 100 documents were retrieved. We downloaded the first 100 documents and ranked them using the term-weight formulas estimated using our techniques. We compared the first 50 documents retrieved by the news search engine and the first 50 documents according to the estimated document similarities. We find that on the average, the top 10 documents predicted by our computations include 84.6% of the top 10 documents retrieved by the WebCrawler news search engine. The corresponding percentages for the top 20, 30, 40 and 50 documents predicted by us are 84, 85.5, 84.8 and 84.5 respectively.

5. FUTURE WORK

We plan to further refine our techniques of determining the mathematical expression of a term-weight component. Our current techniques only consider the text present in a document. We will take into account other factors such as the hyperlinks and pictures appearing in a document. We will also perform extensive experiments on other search engines.

6. REFERENCS

[1] C. Baumgarten. A Probabilistic Model for Distributed Information Retrieval. ACM SIGIR Conference, 1997.

[2] James P. Callan, Zhihong Lu and W. Bruce Croft. Searching Distributed Collections with Inference Networks. ACM SIGIR Conference, 1995.

[3] L. Gravano and H. Garcia-Molina. Generalizing GlOSS to Vecter-Space databases and Broker Hierarchies. VLDB, 1995.

[4] L. Gravano and H. Garcia-Molina. Merging Ranks from Heterogeneous Internet Sources. VLDB, 1997.

[5] K.L. Liu, W. Meng, C. Yu and N. Rishe. Discovery of Similarity Computations on the Internet. Technical report, Dept. of EECS, University of Illinois at Chicago.

[6] W. Meng, K.L. Liu, C. Yu, X. Wang, Y. Chang and N. Rishe. Determining Text Databases to Search in the Internet. VLDB, 1998.

[7] W. Meng, K.L. Liu, C. Yu, W. Wu and N. Rishe. Estimating the Usefulness of Search Engines. ICDE'99.

[8] E. Voorhees, N. Gupta and B. Johnson-Laird. Learning Collecction Fusion Strategies. ACM SIGIR Conference, 1995.

[9] C. Yu and W. Meng. Principles of Database Query Processing for Advanced Applications. Morgan Kaufmann, San Francisco, 1998.