IJCSC International Journal of Computer Science & Communication Vol. 1, No. 2, July-December 2010, pp. 105-108

A Survey on Improving the Web Search Ranking by User Behavior Information

Mohd. Husain[1], Amarjeet Singh[2], Manoj Kumar[3] & Rakesh Ranjan[4]

[1]Department of Computer Science and Engineering, Azad IET, Lucknow
[2,4]Department of Computer Applications, Institute of Environment & Mgmt., Lucknow
[3]Department of Information Technology, IISE, Lucknow
Email: [1]mohd.husain90@gmail.com, [2]amarjeetsingh_9@rediffmail.com, [3]iisemanoj@gmail.com,
[4]rakeshranjan.lko@gmail.com

ABSTRACT

In this paper we propose new methods for ordering the Web pages returned from search engines. Given a few search keywords, nowadays most search engines could retrieve more than a few thousand Web pages. The problem is how to order the retrieved Web pages and then to present the most relevant Web pages first. We propose new factors to allow relevant Web pages to be ranked higher. The factors include keyword popularity, keyword to Web page popularity, and Web page popularity. The keyword to Web page popularity records which Web pages have been selected corresponding to the search keywords. The Web page popularity determines how often the Web pages have been selected and also how many popular keywords are contained in the pages. Using these popularity factors, our system is able to rank more popular pages higher, which will help most search engine users find the more popular and plausibly the more relevant pages.

Keywords: Web Page Ranking, Search Engine, Information Retrieval, Web Mining

1. INTRODUCTION

We are facing information overloaded. Finding information relevant to what we are seeking is becoming more important as the Web is growing in explosive speed. Nowadays, most people try to find whatever information on the Web by using search engines. Given a few search keywords, most search engines today will retrieve more than a few thousand Web pages. The problem now is that we need to scan pages after pages, manually and time consumedly, to find what we need or often give up without getting the needed information. There are several approaches to address the problem. The currently most popular method to address the problem is by ordering the search results and presenting to the users the most relevant pages first. This method is called page ranking, which is one of the important factors that makes Google currently the most successful search engine. Google uses over 100 factors in their methods to rank the search results. Their methods seem to help Web users find the needed information quicker than their competitors. Even with the help of page ranking, we are facing the problem of manually performing sequential search through Web pages after Web pages.

Another approach to help Web users to find the information that they need is by presenting the search results in a hierarchical structure much like a directory tree structure. Using the tree structure, the Web users can browse from one group of Web pages to another group, much like browsing the computer files on a directory tree.

In this paper, we attempt to improve existing page ranking methods. We introduce new factors, which have not been used by Google, to allow relevant Web pages to be ranked higher. We attempt to capture the search history and the preferences of millions of search engine users. Once a user enters a search keyword into our search engine, the keyword is recorded. And, once the user selects a Web page, the URL of that page is recorded. Moreover, the keyword to the URL relation is also recorded. Based on these recorded data, we define three factors, which are keyword popularity, keyword to Web page popularity, and Web page popularity.

Using the popularity factors, our system is able to rank more popular pages higher, which will help most search engine users find the more popular and plausibly the more relevant pages. The idea of relevance is subjective and thus is difficult to be measured. A page relevant to one person may not be relevant to another. Our assumption is that if a page is relevant to a large number of people, it may also be relevant to another person.

2. PRIOR APPROACHES

Since the success of search engine depend on its ranking methods, the research on Web page ranking has received a lot of attention. However, since an effective ranking method has its commercial value, many of the research results have been patented. One the most famous work is the PageRank, which was developed in Stanford

University, patented, and licensed exclusively to Google. Even the name "PageRank" is a trademark of Google. The key idea of the method is to view all the Web pages forming a weighed graph, having each Web page as a vertex and the links between Web pages as the edges. Each Web page is assigned a weight to measure its importance, that is, a Web page, having a large number of other Web pages linking into it, will become more important.

Many researches have focused on extending and improving PageRank method. For instance, the research in Gianna et al (2006) focused on improving the processing speed. The research in Xing and Ghorhani (2004) focused on taking into account the importance of the in-links and out-links and the popularity of Web pages. The research in Shi et al (2003) focused on performing distributed page ranking on top of peer-to-peer networks.

Many new and innovative ideas have been proposed for ranking Web pages. For instance, Diligenti et al (2004) proposed a unified probabilistic framework for ranking Web page. Wu and Aberer (2003) related the behavior of Web surfing to Swarm Intelligent and ranked Web pages based on the interactions of the Web surfers and the search engine. Yuwono and Lee (1996) applied and extended various information retrieval techniques for Web page ranking.

3. DEFINING POPULARITY FACTORS

We define popularity factors that attempt to capture search history and the preferences of millions of search engine users. Currently, Web users interact with search engines by providing several search keywords and selecting Web pages from the search results. We attempt to capture as much usage information as possible and to make use of captured information.

The first factor to be defined is the keyword popularity. When a user entered keywords and clicked search, the search engine will store the keywords and update their weights. Some words called stop words are removed before storing the keywords in the database. For instance, when a user types "department of computer science", the word "of" is not stored as the search key. The order of the words is taking to consideration. For instance, the term "computer science" is store as it is in that order. If a user type "science computer" then a new entry will be create to capture this new terms. Each of the terms, be it a single word or several words, will be associated with a weight that records the frequency that the terms have been used.

The second factor to be defined is the keyword to Web page popularity. After the search engine returns the search results to the user, the user will select Web pages for viewing. The relationships between the search keywords and the selected Web pages will be recorded. The relationships capture the preferences of the users. Some search engines, such as Google, currently cannot capture the relationships. Using Google, for example, when a user clicks on a link on the search results, the browser directly goes to retrieve the Web pages based on the given URL. The search engine does not know what link has been clicked. To allow the search engine to know what link clicked, each click needs to be passed through the search engine. The search keywords and the destination URL is embedded on each link provided on the search results. When a user clicks a link, the browser passes these data to the search engine. The search engine records the data and then redirects the browser to go to retrieve the destination Web page.

The third factor to be defined is the Web page popularity. There are several ways to define the Web page popularity. The most obvious way is to define it as the number of times a Web page has been selected. When a user clicks on a link on the search results, the Web page associated with the link is recorded. This information can be collected when the second factor described above is collected. This method to define Web page popularity should be accompanied by measuring the amount of time a user spent on reading the Web page. This information can be collected by determining the difference between two time stamps of two consecutive clicks. Whenever a user clicks on a link, the time is recorded by the search engine. The assumption is that the user clicks on a link, reads the retrieved Web page, and then clicks on another link.

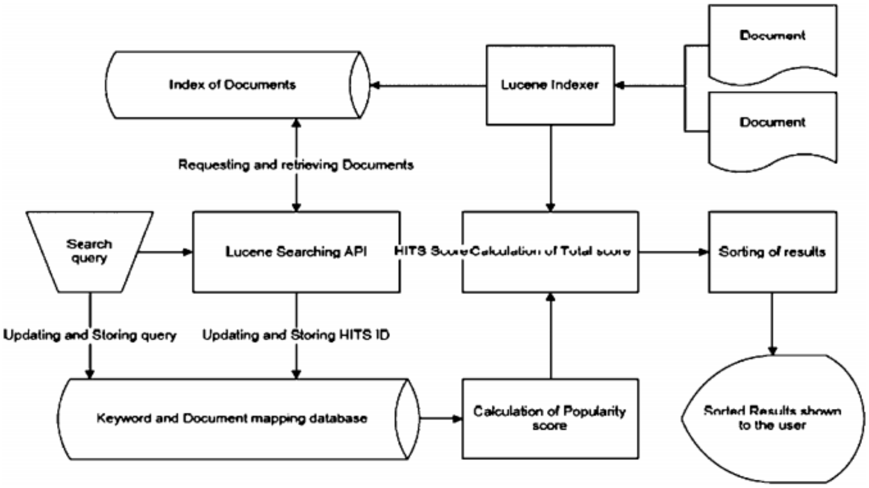
In here, we introduce a new way to define the Web page popularity by counting the number of popular keywords contained in the page. The idea is that if a Web page contains a large number of popular keywords, then it should be considered as more popular. All these ways of defining the Web page popularity can be combined to from a comprehensive one.

4. USING POPULARITY FACTORS

We describe how to make use of the popularity factors for ranking and for improving the usability of search engines. Each of the popularity factors can be used in several ways. The keyword popularity can be used for ranking and for improving the usability of the search engine. One way to use the keyword popularity is to automatically complete the search term entry. After a user keyed in several letters, the search engine automatically retrieves popular keywords associated with the first few letters and shows a list of search terms for the user to choose. Another way to use the keyword popularity is to cache those Web pages associated with some of the most popular keywords such to improve the speed for retrieval. And, as described above, the keyword popularity can also help to define the Web page popularity.

The keyword to Web page popularity can be used for ranking. This keyword to Web page relationship helps the search engine to retrieve popularity pages associated with that particular keyword and to ranks them higher. Our assumption is that if a large number of people searching a particular keyword and preferring some particular Web pages, then these references may also be relevant to another person.

The Web page popularity can also be used for ranking. Again, the idea is to rank more popular pages higher. However, we must address the problem of upward spiral. Since more popular pages are ranked higher, they are more likely to be selected and thus become more popular. One ways to address this problem is to introduce negative factors that reduce the rank of Web pages. Google, for example, uses about one-third of their factors as negative factors.


Document
Document
Index of Documents
Lucene Indexer
Requesting and retrieving Documents
Search query
Lucene Searching API
HITS Score
Calculation of Total score
Sorting of results
Updating and Storing query
Updating and Storing HITS ID
Keyword and Document mapping database
Calculation of Popularity score
Sorted Results shown to the user

System Architecture

5. SYSTEM IMPLEMENTATION AND TESTING

In this section we describe the implementation of a system to test the proposed popularity factors. Instead of building an entire search engine from scratch, we modify an existing open source search engine called Lucene (Apache 2008) for our testing. To store the captured data, we choose to use an object-oriented database called db4o, which is an open-source object database designed to be as simple and fast as possible for Java and .NET software developers. And, we use Java language.

Web pages are first indexed by Lucene indexer and stored in database. The user queries, that are the search keywords, are stored and updated in the database and passed to Lucene search API. The scores for popularity factors are calculated and combined with the ranking scores of Lucene. Then, the retrieved Web pages are sorted according the ranking scores and then represented to the users.

In this process, a user interacts with the testing system in the same as interacting with a search engine. During the interactions, the popularity factors are captured and updated. The score of the popularity factors are combined with the score produced by Lucene. Lucene defines the score of query q for document d, score(q, d) to be as follows (Apache 2008).

$$\text{Score}(q, d) = \text{coord}(q, d)\ .\ \text{queryNorm}(q).\ \Sigma\ (tf(t(n\ d).tdf(z)\ 2.t\ .\ \text{getBoost}().\ \text{norm}(t.d))$$

The key factors in the equation are: the tf(t in d), which is defined as the number of times term *t* (a term *t* is a search keyword in a multiple-keyword query *q*) appears in the document d; the *idf*(*t*), which stands for inverse document frequency and is defined as the number of documents in which the term t appears; and coord (*q*, *d*), which is defined as how many of the query terms are found in the specified document d. We define the score of our popularity factors for search query q and document *d*, pop(*q*, *d*), as shown below:

$$\text{Pop}(q,d) = \text{kpNorm}\ \Sigma\ \text{keyworkPop}\ (t, d) + \text{keywordWebPagePop}(q,d)\ .\ \text{kvpNorm} + \text{WebpagePop(d)}.\ \text{WPNorm}$$

Three of the popularity factors are included in the equation: the keywordPop(*t*,*d*) is the keyword popularity of term t that is part of query q and that term t is contained in document *d*; the keyword WebpagePop(*q*,*d*) is the keyword to Web page popularity factor; and the WebpagePop(*d*) is the Web page popularity of the Web page d. Each the factors are normalized by its corresponding normalization factors and summed together to form the final popularity factor pop (*q*,*d*). The final ranking score is the combination of the score(*q*,*d*) and the pop(*q*,*d*). Each of those scores is normalized before the combination.

Our testing results have shown that each of the popularity factors has effects on making the more popular pages rank higher. We have also adjusted all those normalization factors to determining their effects on ranking. For testing our system, we asked students to use our search engine. During the testing, our system captured and recorded the usage history, which was then translated into the popularity factors. Those popularity factors in combination with other ranking factors are then use to generate new ranking results for future users. We compared our ranking results to Google results by asking new users to enter search keywords in our system and also enter the same keywords in Google. The user justified which search results are relevant to them. We choose "Precision at k" (Agichtein 2006) as our metrics to evaluate ranking relevance. Given a search query, the precision at k, P(k), is defined to be ratio of the number of relevant results contained in the top k results over the value k.

6. CONCLUSION

In this paper we focused on ranking Web pages based on popularity factors, which capture the preferences of millions of users. Three types of popularity factors were defined: the keyword popularity, the keyword to Web page popularity, and the Web page popularity. We described how to collect data for these factors and implemented a system to test the effects of these factors on ranking. Although we were able to come up an equation that allows more popular pages to be ranked higher, we were yet to solve the billion-dollar research problem of finding an optimal equation that can account for a large number of factors and produce the most relevant search results to the users. This is due to the difficult of defining relevance. Some results are relevant to some users under certain conditions but may not be relevant to other users. Our working assumption is that if given certain search keywords and a large number of users prefer certain Web pages, then those Web pages may also be relevant to another user.

REFERENCES

[1] Apache Software Foundation, 2008, "Similarity (Lucene 2008-10-02_02-04-48 API), "http://hudson.zones.apache.org/ hudonjavadoc/org/apache/ lucene/ search/ Similarity.html.

[2] Vaughn, 2008. "Google Search Engine Optimization Information," http://www.vaughns-1-pagers.com/ internet/googleranking-factors.htm.

[3] Diligenti, M., Gori, M., and Maggini, M., 2004, "A Unified Probabilistic Framework for Web page scoring Systems," *IEEE Transactions on Knowledge and Data Engineering*, **16**, Issue 1, pp. 4-16.

[4] Apache Software Foundation, 2008. "Welcome to Lucene", http://lucene.apache.org/.

[5] Db4objects, Inc., 2008. "db4o::Native Java & .NET Open Source Object Database", http://www.db4o.com/.

[6] Yao, Z. and Choi, B., 2007. "Clustering Web Pages into Hierarchical Categories," *International Journal of Intelligent Information Technologies, Special Issue on Web Mining*, **3**, No. 2, pp.17-35.

[7] Agichtein, E., Bill, E., and Dumais, S., 2006. "Improving Web Search Ranking by Incorporating user Behavior Information," *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 19-26.

[8] Gianna M., Corso, D., Gullí, A, and Romani, F., 2006. "Fast PageRank Computation via a Sparse Linear System," *Internet Mathematics*, **2**, No. 3: 251-273.