A Score based Web Page Ranking Algorithm

M. Shamiul Amin
Department of Computer Science and Engineering,
Dhaka International University,
Bangladesh

Shaily Kabir
Department of Computer Science and Engineering,
University of Dhaka,
Bangladesh

Rasel Kabir
Department of Computer Science and Engineering,
University of Dhaka,
Bangladesh

ABSTRACT

With the explosive growth of information in the Web, users face difficulties while finding their desired information. Search engine helps the user by retrieving useful information from this huge collection based on his/her search query and presents a list of relevant web pages as a search result. However, without proper ranking of pages in the result through the relevancy of pages to the search query, the user may need to explore the whole list for discovering the appropriate page(s), thereby involving huge search time. Although a number of ranking algorithms such as HITS, PageRank, Weighted PageRank and etc., are developed to assist the search engine, but none of them provides page ranking with high accuracy. In this paper, we propose a score-based web page ranking algorithm involving web content mining and usage information of the pages. Our algorithm considers both syntactical and semantic matches of the search query to the pages. For a web page, syntactical score is calculated based on the total number of exact matches of the search words in the page. Besides, semantic score is measured using synonym matches of the search words. Moreover, we incorporate the usage information of the pages as page popularity in order to comprise the user interest in the ranking order. The total relevant score of each page is calculated using the summation of the syntactical and semantic scores of the page and its page popularity. Finally, the pages are ranked according to their total relevant score. Based on several performance evaluation measures, experimental results show considerable improvement in the page ranking using our algorithm as compared to other known approaches.

General Terms

Search Engine, Ranking Algorithm, Web content mining, Web usage mining.

Keywords

HITS, PageRank, Score Based Page Rank (SBPR).

1. INTRODUCTION

Nowadays World Wide Web (WWW) is a popular medium to gather information. But web contains huge amount of data. It is really a hard work to find out a specific page or some pages from the huge amount of data. A user may remember some URL of web pages, but it is impossible for any user to remember the URLs of all web pages. Fortunately, in this case search engine works as an agent between the users and WWW. The search engine can retrieve data from WWW according to the user query. It makes easy to find something in WWW. But it is not easy for search engine to search information from the Web and bring back to the user with relevant information in a correct order of page list. To do so, Search engine uses different types of ranking algorithms to order the retrieved pages. Different algorithms use different types of mining patterns. Some use structure mining some use content mining but all of them have some limitations. We will discuss this topic later in this paper.

This paper presents an algorithm, which is based on the web content mining and usage information of pages. First, we retrieve the relevant web page from our database based on user search query. After getting the relevant web pages, we apply our algorithm on those web pages to order the pages.

2. RELATED WORK

Han et al. [1] tried to distinguish the web content mining from two different points. One is Retrieval view and another one is database view. Laplas [2] gave the brief description about different types of web mining and research area of web mining. Chakrabarti et al. [3] described crawler, which explore its crawl boundary to discover the links that are probably to be most pertinent for the crawler. Kobayashi and Takeda [4] discussed the development of new strategies targeted to give solution of some problems such as noise, slack recovery speed and broken links associated with web-based information recovery. Mayfield et al. [5] explored the indexing using both words and N-grams by using a HAIRCUT (Hopkins Automated Information Retrieving for Combing Unstructured Text) system. Kim and Kwon [6] proposed a method of information retrieval using the context information by adopting different types of page ranking algorithms, which are context tags algorithms. Cho et al. [7] described the efficient approach to improve web crawlers by identifying replicated document collections. Brin and Page [8] gave an in-depth description of web search engines of large scale as well as described the PageRank algorithm. The algorithm states that the topicality of a page increases with the number of hyperlinks to it from other topical pages. Beigi et al. [9] introduced MetaSEEkA, a meta-search engine which is based on content, used for discovering images on the Web based on their visual information. MetaSEEkA was designed to penetratingly select and interface with diverse on-line image search engines by ordering their performance of user queries for different classes. Nguyen [10] presented a new web usage mining process for finding sequential patterns in web usage data which can be used for predicting the possible next move in browsing sessions for web personalization. Cooley et al. [11] described WEBMINER, a system for Web usage mining. Ramulu et al. [12] described the over view of semantic mining. Chakravarthy [13] proposed a research on how semantic web technologies can be used to mine the web, for information extraction and examined how new unsupervised processes can aid in extracting precise and useful information from semantic data.

3. WEB PAGE RANKING ALGORITHM

Many web page ranking algorithms are used by different search engines. Here we give the two main web page ranking algorithm with their limitations.

3.1 HITS Algorithm

HITS stands for hypertext induced topic search. This algorithm is mainly link analysis based algorithm. It is developed by Jon Kleinberg [14]. HITS is a query depended algorithm. The main keywords for HITS algorithm are hubs and authorities. Authorities are the central web pages for particular query topics. For wide ranges of topics, the strong authorities consciously do not link with one another [14]. They can only connected by another layer which is known as intermediate layer, the hub pages. A good hub page points to good authorities. On the other hand, a good authority is pointed by good hubs. Relationship between hubs and authorities is the hub weight to be the sum of the authorities of the nodes that are pointed to by the hub, and the authority weight to be the sum of the hub weights that point to this authority [14]. HITS algorithm is based on two equations which are shown in Eq. (1) and (2).

$$a_p = \sum_{q,q->p} h_q \qquad (1)$$

$$h_p = \sum_{p,q->q} a_q \qquad (2)$$

From Eq. (1) we can find that a page p's authority is updated by the sum of the of hub pages which is denoted by q. On the other hand, in Eq. (2) a page's hub is updated by the sum of the authority's page which is denoted by q. However, there are two major limitations of HITS algorithm [15]. One limitation is mutually reinforcing relationships between hosts and other one is topic drift. An attempt to solve these problems is to add weights to web documents. Moreover, software *"Linkviewer"* designed in [16] had shown that in some cases HITS algorithm gives poor result. When they gave *"Harvard"* as a query, they got Havard home page (*http://www.harvard.edu/*) as 67th authority.

3.2 PageRank Algorithm

It is also a link analysis algorithm. This algorithm is used by Google web search engine. PageRank algorithm is formulated by Brin and Page [17] in their paper *"The Anatomy of a Large-Scale Hyper Textual Web Search Engine"*. Unlike the HITS algorithm, PageRank is a query independent algorithm. The rank score of each page is determined by the link with those pages. Therefore, importance of a page is increased by increasing the number of back link. Here, the rank of a page P is calculated using Eq. (3).

$$r(p) = \sum_{Q \in B_p} r(Q) / |Q| \qquad (3)$$

Here, B is all page linking to P. | Q | is the number of outlinks from the page Q. In Eq. (4) we rewrite the equation in an optimal way [18].

$$PR(A) = \frac{(1-d)}{N} + d(\frac{PR(T_1)}{C(T_1)} + ... + \frac{PR(T_N)}{C(T_N)}) \qquad (4)$$

Where PR(A) stands for the PageRank of page A, PR(T_i) stands for the PageRank of pages T_i which link to page A, C(T_i) stands for the number of outbound links on page T_i,, d stands for damping factor which can be set between 0 and 1 and N is the number of web pages [18]. In this algorithm, the rank of page A is recursively defined by the ranks of those pages which are linked to page A.

However, there are some major limitations of PageRank algorithm [19]. A common problem is spider traps. If there are no links from within the group to outside the groups then the group of pages is a spider traps. Another problem is the rank sink problem. It occurs when a network of pages falls in an infinite cycle. Moreover, there exists dead-ends and dangling, links problem. Dead-ends problem occurs, when pages have no out links. Sometimes a page contains a link of another page which has no out links. These types of link are known as dangling links [19].

4. OUR APPROACH

Our proposed Score Based Page Ranking (*SBPR*) is based on the content mining as well as the usage information of pages. We have calculated a score for each page which is related to query using the frequencies of keywords of that query as well as the synonyms of those keywords, and also using the popularity of that specific page. In our algorithm each page gets as core for that reason our algorithm is known as *"Score Based Page Ranking Algorithm"*. The main equations of our approach are:

$$WPRS_i = \sum_{S=0}^{S=N} (F_S * C_S) + PP_i \qquad (5)$$

$$C_S = C_{s-1} * d \qquad (6)$$

Where, $WPRS_i$ is Web Page Relevancy score of page P_i , F_S is frequency of each keyword S, C_S is the emphasis factor of each keyword S, PP_i is Page popularity of page $P_{i,,}$ d is dumping factor, N is Number of keywords contained in the user query. We have considered both the syntactical as well as the semantic matching of keywords to the user search query, for semantic matching, we have used those synonyms which are popular to the users. We use software *"WordNet"* to find synonym words for specific keyword. *"WordNet"* is a large lexical database for the English language. It groups English words into sets of synsets which are usually synonyms, records the several semantic relations between these synonym sets [20]. Rather than give same priority to every keyword of a specific query we give different priority to different keyword. For this reason, we have added the emphasis factor among the keywords of given query. We use emphasis according to the order. That is, the first keyword of query gets the higher priority than second keyword; the second keyword gets higher priority than third keyword and so on. Dumping factor is a very important factor in Google Pagerank algorithm. Calculation may change based on changing the dumping factor. In our algorithm, dumping factor is responsible for changing the value of emphasis factor. From Eq. (6) we can see that emphasis factor of a keyword is based on the emphasis factor of previous keyword and the dumping factor. Here, the value range of dumping factor is greater than 0 and less than 1.

4.1 Algorithm

Input : A user query with search word SW=S_0, S_1,...,S_n; SW is a list of search words, WPRS is Web page relevant score, WPI is web page information, C is emphasis factor, d is dumping factor, PP is page popularity, LIST is list of pages, t is the number of pages contain a specific keyword.

Output: A ranking list of web pages.

```
Initialize WPRS=0
Set C=0.1
for $i = 0$ to $n$ do
  For each page $P_k$ with $S_i \subset$ WPI do
    if $P_k$ ε LIST then
      insert $P_k$ in LIST
    end if
    for j = 0 to t do
      Find frequency $F_i$ of $S_i$ for each page $P_j$ ε LIST
      Calculate
        $WPR_j = WPRS_j + (F_i \times C_i)$
    end for
    $C_i = C_{i-1} \times d$
  end for
if LIST contains l Pages then
  for k=0 to l do
    Find the popularity $PP_k$ of page $P_k$
    $WPRS_k = WPRS_k + PP_k$
  end for
end if
Rank the LIST according to their WPRS score
return LIST
```

5. EVALUATION

We have created a database to evaluate our proposed *SBPR* algorithm. We have selected Google custom search engines algorithm for comparing with our algorithm because we can use same database for it.

5.1 Impact of Page Popularity

Page popularity is very important feature in our *SBPR* algorithm. Using this page popularity we can get the dynamic scoring of a specific web page for a specific query. Suppose, our search word "*java*". We have given first ten links with their score for this keyword in Figure 1. From Figure 1, we can see that the link "*http//:www.java.com/*" has get 9th in the list. However, it may get higher ranking position in the rank list if this page becomes popular by the users. A web page becomes popular if the page gets many users to visit.

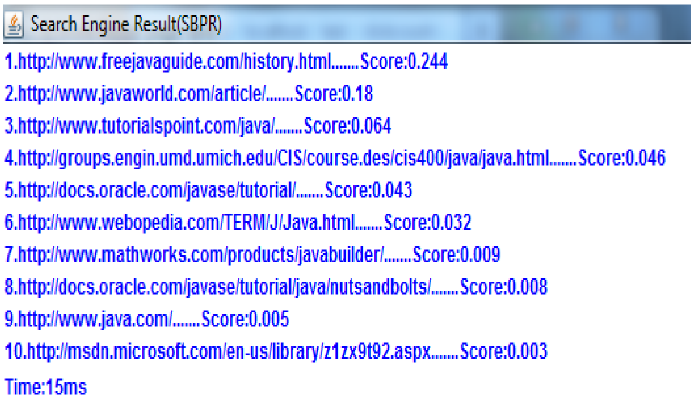
Suppose, many users have visited the link *"http//:www.java.com/"*, therefore the page popularity of that page is increased. In Figure 2, we can see that the page popularity of the link *"http//:www.java.com/"* is increased and it gets the first position in the rank list.

5.2 Precision

Precision is an important field for performance analysis. Precision denotes the proportion of enumerated positive cases that are accurately actual positives [21]. There is a equation for calculating precision which is shown in Eq. (7).

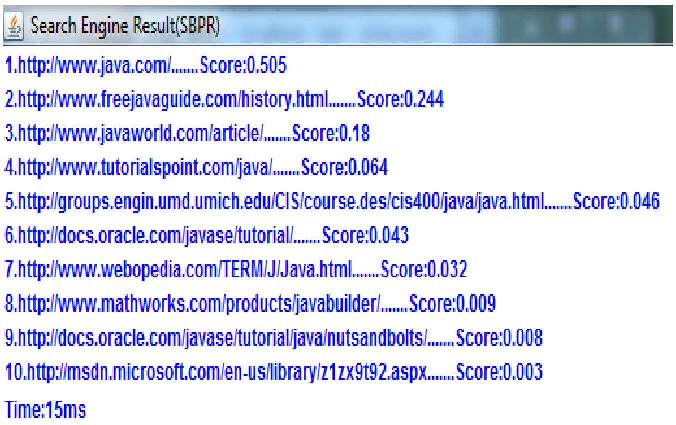
$$precision = t_p /(t_p + f_p) \qquad (7)$$

Where,
t_p= true positive (correct result).
f_p=false positive (unexpected result).



Search Engine Result(SBPR)
1.http://www.freejavaguide.com/history.html.......Score:0.244
2.http://www.javaworld.com/article/.......Score:0.18
3.http://www.tutorialspoint.com/java/.......Score:0.064
4.http://groups.engin.umd.umich.edu/CIS/course.des/cis400/java/java.html.......Score:0.046
5.http://docs.oracle.com/javase/tutorial/.......Score:0.043
6.http://www.webopedia.com/TERM/J/Java.html.......Score:0.032
7.http://www.mathworks.com/products/javabuilder/.......Score:0.009
8.http://docs.oracle.com/javase/tutorial/java/nutsandbolts/.......Score:0.008
9.http://www.java.com/.......Score:0.005
10.http://msdn.microsoft.com/en-us/library/z1zx9t92.aspx.......Score:0.003
Time:15ms

Fig 1: Ranking order of *SBPR* for query *"java"* without page popularity



Search Engine Result(SBPR)
1.http://www.java.com/.......Score:0.505
2.http://www.freejavaguide.com/history.html.......Score:0.244
3.http://www.javaworld.com/article/.......Score:0.18
4.http://www.tutorialspoint.com/java/.......Score:0.064
5.http://groups.engin.umd.umich.edu/CIS/course.des/cis400/java/java.html.......Score:0.046
6.http://docs.oracle.com/javase/tutorial/.......Score:0.043
7.http://www.webopedia.com/TERM/J/Java.html.......Score:0.032
8.http://www.mathworks.com/products/javabuilder/.......Score:0.009
9.http://docs.oracle.com/javase/tutorial/java/nutsandbolts/.......Score:0.008
10.http://msdn.microsoft.com/en-us/library/z1zx9t92.aspx.......Score:0.003
Time:15ms

Fig 2: Ranking order of *SBPR* for query *"java"* with page popularity

We have given a table which contains the precision of top ten link of the rank list for our algorithm SBPR as well as for Google custom search algorithm.

Table 1. Precision for different queries

Query	*SBPR*	Google search engine algorithm
Java	1.0	1.0
dhaka	1.0	1.0
rent a car	1.0	1.0
car	1.0	1.0
Web mining	1.0	1.0
Dhaka city	1.0	1.0
Car rent	1.0	1.0
Web content mining	1.0	1.0
Computer virus	1.0	1.0

From Table 1, we can see that for first ten links of rank list, our algorithm and Google custom search gives same precisions.

5.3 Recall

Recall is another important field for performance evaluation. The equation of calculating recall is shown in Eq. (8).

$$recall = t_p /(t_p + f_n) \qquad (8)$$

Where,
t_p= true positive (correct result).
f_n = false negative (missing result).

In Figure 3, we have given a graph for recall of ten different queries based on first ten link of rank list for our *SBPR* algorithm as well as Google custom search algorithm. When we try the “rent a car” or “car rent” as search word we get poor recall for Google custom search.

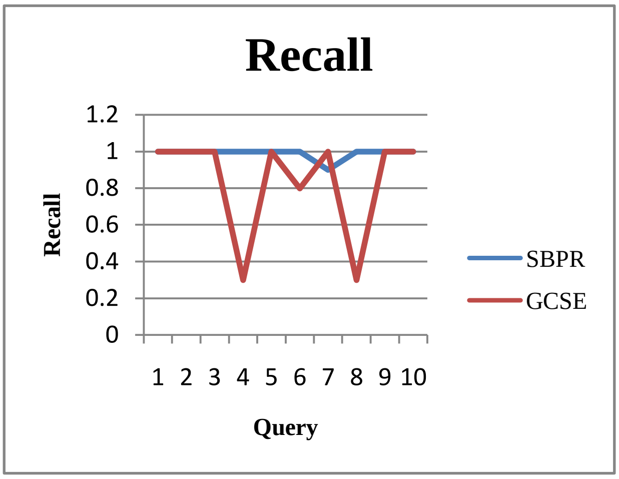


Fig 3: Recall

5.4 F-measure

F-measure is a measure of test accuracy. It considers both the precision and the recall of the test to compute the score. The equation of calculating is shown in Eq. (9).

$$fmeasure = \frac{2(precision \times recall)}{precision + recall} \qquad (9)$$

In Figure 4, we have given a graph for f-measure of ten different queries based on first ten link of rank list for our *SBPR* algorithm as well as Google custom search algorithm.

From Figure 3 and Figure 4, we can say that our SBPR algorithm is working better than Google custom search algorithm in many cases.

6. CONCLUSIONS

In our approach, we have used two types of mining techniques, content mining and usage mining. If we could add structure mining in our algorithm, we may get better result. Therefore, in future we will incorporate the structure mining in our approach. For page popularity factor we have considered the number of visits of users for a specific page; however it is also important that how much time a user spends on specific page. We hope we can add this feature of usage information in our algorithm near future for calculating the page popularity in a more precise way.

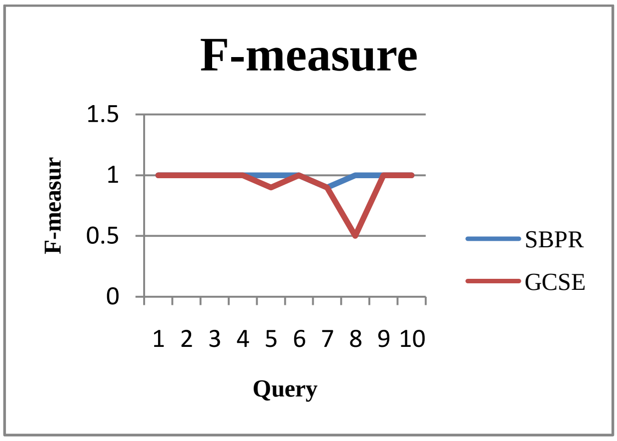


Fig 4: F-measure

7. REFERENCES

[1] Han, J., Kamber, M. (2006). Data Mining, Southeast Asia Edition: Concepts and Techniques. Morgan kaufmann.

[2] Lappas, G. (2008). An overview of web mining in societal benefit areas. Online Information Review, 32(2), 179-195.

[3] Chakrabarti, S., Van den Berg, M., Dom, B. (1999). Focused crawling: a new approach to topic-specific Web resource discovery. Computer Networks, 31(11), 1623-1640.

[4] Kobayashi, M., Takeda, K. (2000). Information retrieval on the web. ACM Computing Surveys (CSUR), 32(2), 144-173.

[5] Mayfield, J., McNamee, P., Piatko, C. D. (1999). The JHU/APL HAIRCUT System at TREC-8. In *TREC.*

[6] Kim, S., Kwon, J. (2007). Effective context-aware recommendation on the semantic web. International Journal of Computer Science and Network Security, 7(8), 154-159.

[7] Cho, J., Garcia-Molina, H., Page, L. (1998). Efficient crawling through URL ordering. Computer Networks and ISDN Systems, 30(1), 161-172.

[8] Brin, S., Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. Computer networks and ISDN systems, 30(1), 107-117.

[9] Beigi, M., Benitez, A. B., Chang, S. F. (1997, December). MetaSEEk: a content-based metasearch engine for images. In Photonics West'98 Electronic Imaging (pp. 118-128). International Society for Optics and Photonics.

[10] Nguyen, S. T. (2009, December). Efficient web usage mining process for sequential patterns. In Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services (pp. 465-469). ACM. Cooley, R., Mobasher, B., Srivastava, J. (1997, November). Web mining: Information and pattern discovery on the world wide web. In Tools with Artificial Intelligence, 1997. Proceedings. Ninth IEEE International Conference on (pp. 558-567). IEEE.

[11] Ramulu, V. S., Kumar, C. N. S., Reddy, K. S. A Study of Semantic Web Mining: Integrating Domain Knowledge

into Web Mining. International Journal of Soft Computing and Engineering (IJSCE), 2(3).

[12] Chakravarthy, A. (2005). Mining the Semantic Web.

[13] Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. Journal of the ACM (JACM), 46(5), 604-632.

[14] Bharat, K., Henzinger, M. R. (1998, August). Improved algorithms for topic distillation in a hyperlinked environment. In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (pp. 104-111). ACM.

[15] Nomura, S., Oyama, S., Hayamizu, T., Ishida, T. (2004). Analysis and improvement of HITS algorithm for detecting Web communities. *Systems and Computers in Japan*, *35*(13), 32-42.

[16] Page, L., Brin, S., Motwani, R., Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web.

[17] PageRank algorithm. http://pr.efactory.de/e-pagerank-algorithm.shtml. Accessed: 2014-04-15.

[18] Karpeles M. 2009. Modeling and optimizing hyper textual search engines, based on the research of Larry page and sergey brin. Yunfei Zhao Department of Computer Science, University of Vermon Slide from spring.

[19] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database*. International journal of lexicography, 3(4), 235-244.

[20] Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.