Advances in Computer Science and Information Technology (ACSIT)
Print ISSN: 2393-9907; Online ISSN: 2393-9915; Volume 1, Number 2; November, 2014 pp. 90-93

http://www.krishisanskriti.org/acsit.html

Proposed Methodology for Average Response Time Page Rank Algorithm

Nandnee Jain[1] and Upendra Dwivedi[2]

[1]Computer Science and Engineering Shri Vaishnav Institute of Tec. And Sc. Indore, India
[2]Shri Vaishnav Institute of Tec. and Sc Indore, India
E-mail: [1]nandneejain@yahoo.com, [2]ud1985@gmail.com

Abstract: *World of today is full of electronic objects. WWW is also an important part of electronic objects which deals from ecommerce to knowledge source. Search engine is heart of WWW, So as Page Rank algorithms for any search engine. In the series of wonderful page rank algorithms one more add up is Average Response Time page rank algorithm which has its specialty of using user behavior as well, while ranking web pages. Average Response Time page rank algorithm along with WSM also takes into consideration time spend by user on web page for assigning priority to pages with respect to query fired on search engine. Each time when user submits query to search engine a set of web pages are returned in answer. User visit the pages and it spend more minutes on the web pages which he/she finds worth reading. Time spend is noted and an average is calculated of the previous average time spend and current time to decide the rank of the web page for next time. This algorithm is dynamic in nature due to its time calculation process. Here in this paper inside view of proposed algorithm is mentioned.*

1. INTRODUCTION

Explosion in size of WWW increased difficulties for search engine as millions of queries and billions of pages are to be handled. If search engine is compared with machine than Page Rank algorithms are those parts of a machine which determine success or failure. The most important factors on which search engine judge upon is quantity, quality and arrangement of pages which are output to input queries.

Page Rank algorithm has its root in Web Mining. Web mining is not an individual term but a group of three i.e. Web Content Mining(extraction of information depending on intra structure), Web Structure Mining(extraction of information depending on inter structure), Web Usage Mining(extraction of information using user behavior)[1].Ranking Algorithm uses single concept of Web Mining , combination of two or three of them.

Response Time based page rank algorithm depends upon all three of them. Web Content Mining use to generate the root set i.e. relevant pages on the basis of occurrence of query terms. Web Structure Miner has two roles to play. First one is to generate extended sets of web pages by following the links from root set. Secondly, In assigning priority to the pages. Web Usage Mining for generating time pattern from user behavior. In this algorithm combination of WSM and WUM is used to rank the web pages.

2. OVERVIEW OF SEARCH ENGINE

Consequence of a simple query on a search engine is return of hundreds to thousands to millions of pages. But user of search engine doesn't have much time to go through all the results to find interesting one .According to studies [2,3] , its rare chances that user visit beyond first page of results to satisfy its hunt of information. Basically task of search engine is divided into three parts[4]:

- Selecting the part of web to be crawled depending on the words.
- Generating forward index(parsing document to words)
- Providing solution to queries.

2.1 Working of Search Engine

Extraction of web pages after submission of query by user is multiple step process[4].

URL Server

It act as a feeder to crawler by providing the list of URLS to be fetched.

Web Crawler

A special software robot called spider automatically traverse the links, download the pages and follow the links from one page to another. The main function of crawler is to provide data for indexing. Web Spider like actual spider crawls on web by following the links. Whenever a new URL or link is encountered than it is associated with ID number called docID.

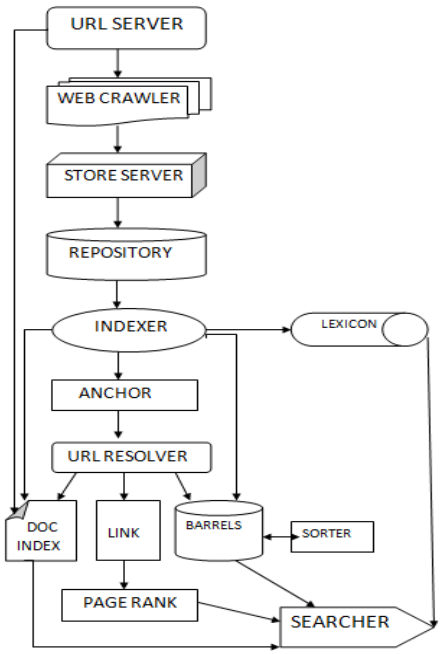

Fig. 1: Search Engine

Store Server and Repository

Store Server receives fetched pages from crawler compress it and send it to repository.

Indexer

Basically indexer performs two functions. Firstly, it visit the repository, uncompress the pages and parses them into words. The parsing of documents into words is called hits. Hits along with word also note its position ,font size and Capitalization and then after they are distributed into barrels. Secondly, indexer along with words also parses out links on documents and store information of links about where link points to and from in anchor file.

Anchors

This file contain links and its information about where it points and from which it is pointed.

URL Resolver

Take input as relative URLs from anchor file and generate output as absolute URLs. After that it assigns Doc Id to URLs.

Sorter

Sorter read the barrels and resorts them to generate inverted index according to word ID.

Lexicon

It takes the lexicon produced by indexer and generates new lexicon for Researchers.

Searchers uses the new lexicon along with inverted index and page rank to answer the queries

Issues Faced by Search Engine

Search Engine is very complex in itself. And again maintaining its performance and quality arises many Issues that to be handled[5]:

- Content Quality
- Vaguely Structured Data
- Spam
- Web Convention
- Quality Evaluation
- Duplicate Host

3. PREVIOUS WORK ON PAGE RANK ALGORITHM

Search engine success doesn't depends only on the result generation but also on the arrangements of result such that top most results should be very useful regarding the topic. For making information easily accessible to user ranking algorithm has been employed. Some of the existing algorithm are described below:

3.1 Page Rank Algorithm

In [8-10], Detailed working of page rank is revealed. This algorithm found its practical implementation in most powerful search engine i.e. Google.

Page rank algorithm is based on web structure miner. It takes inlinks to assign the priority to the pages .Rank of pages can be increased in two cases, firstly when it is pointed by two many pages or few pages of high rank. Calculation of priority is done offline. Equation for page rank algorithm is given by

$PR(a) = (1-d) + d(PR(B_1)/C(B_1) + \ldots\ldots\ldots\ldots + PR(B_n)/C(B_n))$

Where

- PR(a)= page rank of page a,
- $PR(B_i)$=page rank of pages B_i which link to page a,
- $C(B_i)$=outbound links on page B_i,
- D= damping factor whose value is between 0 and 1.Usually set to 0.85.

The rank of each page is divided among the links which moves out of pages and rank of page is summation of all link weight from which it is pointed.

Page rank algorithm suffers from "richer get richer" problem [12]. Current page rank algorithm uses relevancy and importance or quality interchangeably. But quality is calculated at crawl time and relevancy can be calculated after query is framed by user.

Secondly, current popularity is basis on which priority of page is decided in page rank algorithm i.e. currently popular page is returned on the top. User has access to this pages making them more popular which results that some unpopular pages having good are always away from limelight.

3.2 HITS

It is developed by Kleinberg [13]. HITS Become the heart of Clever System[14] .Clever System has introduces some modification in original HITS algorithm[15].

HITS algorithm moves around two concepts Authority and Hub.

- Authority are those pages which are pointed by many hubs and are important according to the topic. They are non self descriptive and can be discovered using hubs [16].
- Hubs are path which can take us to authority. This are the pages which are pointed by many authority[16].

HITS suffer from two problems mutual reinforcement and topic dirft[17].

4. RESPONSE TIME BASED PAGE RANK ALGORITHM

User satisfaction is most important responsibility for any search engine. But in web environment its hard to detect percentage till which user is satisfied as he is not interested in providing the external feedback about the pages. So some mechanism should be devised through which an internal feedback collection can be done without bothering the user. Internal collection is possible only through web content mining. In [6], a new ranking algorithm which takes into account this internal feedback .The average response time page rank algorithm is dynamic in nature. Its an iterative process. When user submits the query through query interface, than root sets are generated using query terms presents on the document. Than after using to and fro links to the pages present in root set base set is generated. Graph of the web is generated using structure miner where nodes represents pages and edges represents the link between pages[7].

Generated web graph is one input for ranking the pages. In beginning , when no time records exist for any pages ranking will be provided depending on link structure or we can according to usual page rank algorithm[9].

When time statistics are generated then both link structure and time statistics become the input to ranking model whose output is different ranking value to different pages. In one sitting an user visit approx. twenty two pages from five web sites[8].So time statistics doesn't exist for whole group of pages than remaining pages which have no time statistics their rank depend fully on page rank algorithm.

After user visit the page than time is recorded that he/she spend on the web pages. Current time and previous average calculated are used to generate new average time statistic. Further this average used next time to decide the priority of page rank algorithm.

5. PROPOSED METHODOLOGY

In Previous paper [6], detailed methodology was not present and some points are untouched which are presented over here. The proposed methodology consists of following phases:

1. User submits the query.
2. On the basis of submitted query a number of URL retrieves.
3. Compute access time of each of the visited url web page.
4. Create web log data for each of the visited web page with their respective access time.
5. Compare the access time with the minimum and maximum threshold access time. Maximum threshold is decided due to reason that when user find something interesting that its starts reading spend lot time or increase priority of time just opening the page and leave it for unlimited duration of time. Secondly, minimum threshold comes to picture as some time after clicking in URL and going spending some seconds he comes to know that page is waste for him.
6. If the access time is less than min threshold time set the access time to zero.
7. Compute page ranking based on the access time of each web page.

Table 1: Notations description

Notations Used	Description
Ui	User
Qi	Query for web page
Tmax	Maximum threshold access Time
Tmin	Minimum threshold Access time
AccTime	Access Time of the page
Avg	Average response Time
PR	Ranking of Web Page
Urli	Retrieved Url's

Notational representation:

1. $U_i \rightarrow Q_i$
2. $Q_i \rightarrow Url_i$

3. Initialize AccTime =0 for each Url_i
4. For i=1: lengh(Url_i)
5. AccTime = AccTime +AccTime(Url_i)
6. if AccTime > T_{max}
7. AccTime= T_{max}
8. Else if AccTime < T_{min}
9. AccTime=0
10. $PR(Url_i) = (1-d) + d* (PR(Url_i)/C(Url_i)$
11. If if avg(AccTime) is incremented from say 15 Seconds (this duration can be changed as per requirement) than
12. $PR(Url_i) = PR(Url_i) + 0.1$
13. END

6. CONCLUSION

Retrieving the required information from the ocean of web is very hard core problem. Different ranking algorithm utilizes different techniques so that useful information can be easily available to the user. Response Time based algorithm is a new turn in the path of page rank algorithm. In these user behavior is kept forefront while deciding the rank of web pages resulting in more satisfaction from users point of view.

REFERENCES

[1] R. Kosala and H. Blockeel. *Web mining research: A survey*.ACM SIGKDD Explorations, 2(1):1–15, 2000

[2] A.Broder, *Web Searching Technology Overview*, Advanced school and Workshop on Models and Algorithms for the World Wide web, 2002.

[3] B.J.Jansen, A.Spink, J.Bateman, T.Saracevic, *Real life Information Retrieval: A study of user queries on the we*b, ACM SIGIR Forum, 1998.

[4] Marcus, Monica Peshave Kamyar Dezhgosha, "*How Search Engines Work And A Web Crawler*", Application, University of Illinois at Springfield, IL 62703,1-15, 2002

[5] S Monika R. Henzinger, Rajeev Motwani, Craig Silverstein," *Challenges in Web Search Engines*"

[6] Nandnee Jain, Upendra Dwivedi , *Average Response Time Based Page Rank Algorithm* , Proceedings of International conference on emerging trends in electronics, electrical and computing technologies,ISBN No. 978-93-5196-068-3 .

[7] Broder A, Kumar R, Maghoul F, Raghavan P, Rajagopalan S,Stata R, Tomkins A, Wiener J ,*Graph structure in the Web*.Comput Netw 33(1–6):309–320 .

[8] Feng Qiu, Zhenyu Liu, Junghoo Cho. Analysis of User Web Traffic with a Focus on Search Activities. International workshop on web and database, 2005

[9] S. Brin, L. Page. *The anatomy of a large-scale Hypertextual Web search engine.* Computer Networks and ISDN Systems, 30, 1998

[10] L. Page, S. Brin, R. Motwani, and T. Winograd, "*The PageRank Citation Ranking: Bringing Order to the Web*", Technical Report, Stanford Digital Libraries SIDL-WP-1999-0120, 1999.

[11] C. Ridings and M. Shishigin, "*Pagerank Uncovered*", Technical Report, 2002.

[12] Cho, J., Roy, S., & E. Adams, R. (2005)." *Page Quality: In Search of an Unbiased Web Ranking".* In Proceedings of ACM International Conference on Management of Data (SIGMOD).

[13] J. Kleinberg. "*Authoritative sources in a hyperlinked Environment".* In Proc. Ninth Ann. ACM-SIAM Symp. Discrete Algorithms, pages 668 {677, ACM Press, New York, 1998

[14] http://www.research.ibm.com/topics/popups/innovate/hci/html/clever.html

[15] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins .*Core Algorithms in the CLEVER System.* ACM Transactions on Internet Technology, Vol. 6, No. 2, May 2006, Pages 131–152

[16] S. Chakrabarti, B. E. Dom, S. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. Kleinberg. *Mining the Web's link structure*. Computer, 32(8):60–67, 1999

[17] K. Bharat and M. R. Henzinger. *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, pages 104{111, 1998.