International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 4, Number 15 (2014), pp. 1575-1581

http://www. irphouse.com

An Adaptive Approach in Web Search Algorithm

Poonam Rawat[1], Shri Prakash Dwivedi[2], Haridwari Lal Mandoria[3]

[1, 2, 3]Department of IT,
Govind Ballabh Pant University of Agriculture & Tehnology,
Pantnagar, Uttrakhand

Abstract

As the information on the World Wide Web is growing every day, user searching for information can be easily lost in the hyperlinked structure of the web. The main goal of the search engine is to return relevant information to the user in respond of a query. In this paper we describe PageRank algorithm and simulate it in PageRank simulator, then we show how text search affects PageRank result and finally with the help of graph PageRank and final PageRank values are compared.

Keywords: PageRank, web search engine, PageRank simulator.

1. Introduction:

Web is huge, structure of the web is hyperlinked and pages on the web semi structured all these are the challenge in retrieving useful information from the web [2, 4]. And as the volume of the web is growing every day it will become more challenging to retrieve of the information. For this purpose we use information retrieval tools like: desktop search engine, digital libraries and web search engine. Web search engine like Google, Bing, Yahoo, Msn etc are use to search information from the web [2]. All these web search engines answer trillions of queries in a day. Web search engine play vital role in search on internet. So first of all we have to understand the basic architecture of the web search engine. Fig. 1 shows the working of a typical search engine.

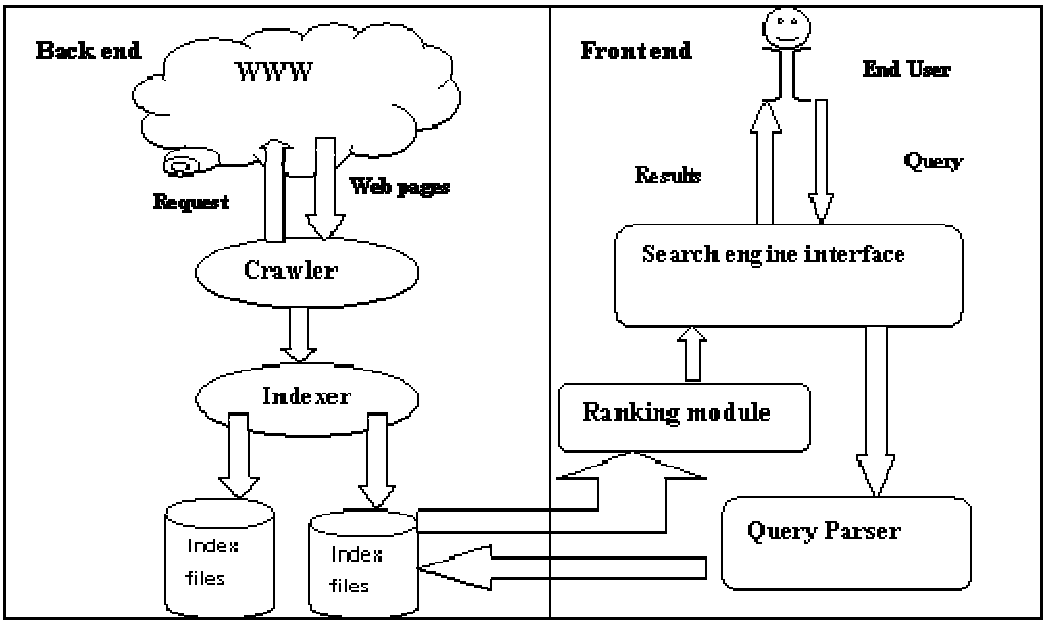

Fig. 1. Web Search Engine Architecture

In the back end of the web search engine we have crawler, indexer and index files. While in a front end we have search engine interface, query processor and ranking module. Crawler crawl the web via recursively visiting web pages through links between web pages then downloads the pages. Indexer extracts the keywords from the downloaded web pages and analyzed them. Then indexer builds index files having a table of keywords and their corresponding web pages. In the front end when user types a query in a search engine interface, query processor analyzes the query and breaks it into keywords. Then match the query keywords with index files keywords and then it returns a list of web pages. And finally ranking mechanism is done by ranking all the return web pages according to chosen ranking algorithm.

2. Ranking mechanism:

Ranking module play an essential role in web search engine, it is used by search engine for ranking search result. For a user queries it determines the order of the return pages in result. Generally the order of the web pages is depends on Popularity of the pages. Web pages' having high Popularity comes up in the returned result. Popularity is also known as Pagerank. Search engine's Results for a query should be arranged on descending order of Pageanks. Pagerank calculation is very critical part of search engine. For this purpose search engine uses page ranking algorithms. Basically page ranking algorithm are use to present the search result by considering the relevance, importance and content score [2]. Some of the most popular page ranking algorithms are PageRank algorithm, weighted Page Rank algorithm and Hyperlink-Induced Topic Search (HITS) algorithm [3].

2. 1. PageRank Algorithm:

In 1996 at Stanford University Larry page and Surgey Brin developed a link analysis algorithm which is known as PageRank algorithm [6]. PageRank algorithm named PageRank after Larry Page, who is co founder of Google search engine. Google is very popular because of its PageRank algorithm. PageRank is Google's method for calculating importance of web pages in web by using the hyperlinked structure of web pages. Suppose we have two pages A, B and page A links to Page B. It means that Page A is saying that Page B is important. If page A links to Page B than we will say that Page A have forward link to Page B and Page B have backlink from Page B. Pagerank algorithm states that if a web page has some important backlinks to it than its outgoing links to other web pages also become important [3]. A simplified version of PageRank is shown in equation 1.

$$PR(A) = \left(\frac{PR(T_1)}{C(T_1)} + \frac{PR(T_2)}{C(T_2)} + \cdots + \frac{PR(T_n)}{C(T_n)}\right)$$

Where PR(A) is the PageRank of Page A, $PR(T_1)$ is the PageRank of a site having back link to Page A, $C(T_1)$ is the number of forward links from that page And $PR(T_n)/C(T_n)$ is calculated for each page which have backlink of page A. PageRank formula is recursive. To calculate PageRank of a page we need to know the PageRank of other pages. Therefore we started with any random value (normally with 1 for all pages) of PageRank and iteratively update these ranks using above formula and wait until they converge [5]. But there is a problem with this simplified ranking function, suppose we have two pages that point to each other but there is no outgoing links from this loop and there are some pages which have incoming links to one of them, then during iteration this loop will accumulate rank but will never distribute any rank, since there are no outgoing link. The loop forms a sort of trap which is known rank sink [1]. To overcome this problem a modification was made on PageRank formula. Modified Page Rank formula is:

$$PR(A) = (1 - d) + d\left(\frac{PR(T_1)}{C(T_1)} + \frac{PR(T_2)}{C(T_2)} + \cdots + \frac{PR(T_n)}{C(T_n)}\right)$$

Where d is damping factor and its value can be set between 0 and 1, we usually set the value of d to 0. 85. The damping factor is use to stop the other pages having too much influence.

3. Proposed Solution:

In our approach we are using Hypertext Markup Language (HTML) pages and rank these pages on the basis of the following web pages or HTML pages parameters:

- PageRank: Page Ranks of the pages are based on the hyperlinked structure of the web pages. We are calculating PageRank by using PageRank Simulator (PageRank sim). In PageRank sim web pages are shown as rectangular boxes and links between them is shown as directional edges.

- Title name: The most important on-page factor is <TITLE> tag. Keywords on title name of web page are more valuable keywords than anywhere else on the page. If a page having desired keyword on its title then it will be more relevant than other pages which are not having desired keyword on their title because if keyword appears in the title then it appears in the content of pages at a high frequency.
- Filename: Importance of filename is similar to the title name.
- Bold and italic words: some special keywords are written in bold or italic form.
- Keyword density: A theoretical desirable ratio of the number of times your keywords occurs in a page to the total number of words on the page.
- Keyword location: if keyword placed near the top of the page than it carry more weight than words on bottom of the pages.
- Heading: heading tags are use to identify the different sections of a HTML page. In HTML pages we have <h1>, <h2>, <h3>, <h4>, and <h5> heading tags.

In this research we have assigned some weighted value to these parameter according to their importance, which are shown in Table 1.

Table 1

Sr. No.	Parameter Name	Weighted value
1	PageRank	As calculated
2	Title name	0. 05
3	File name	0. 05
4	Heading(h1, h2, h3, h4, h5)	0. 03
5	Bold and italic	0. 02
6	Keword density	As calculated
7	Keyword on top 30 lines	0. 01

4. Implementation and Results:

We have implemented our project ‘My Search’ using C# as a language in Visual Studio 2012, SQL Server as a database and PageRank Simulator. We have a collection of web pages in which we have performed search for a query using ‘My Search’. First of all we have created a hyperlink structure of all collected web pages in PageRank sim (shown in fig. 2) then PageRank sim have given us PageRank of all web pages by calculating it with the dumping factor=0. 85.

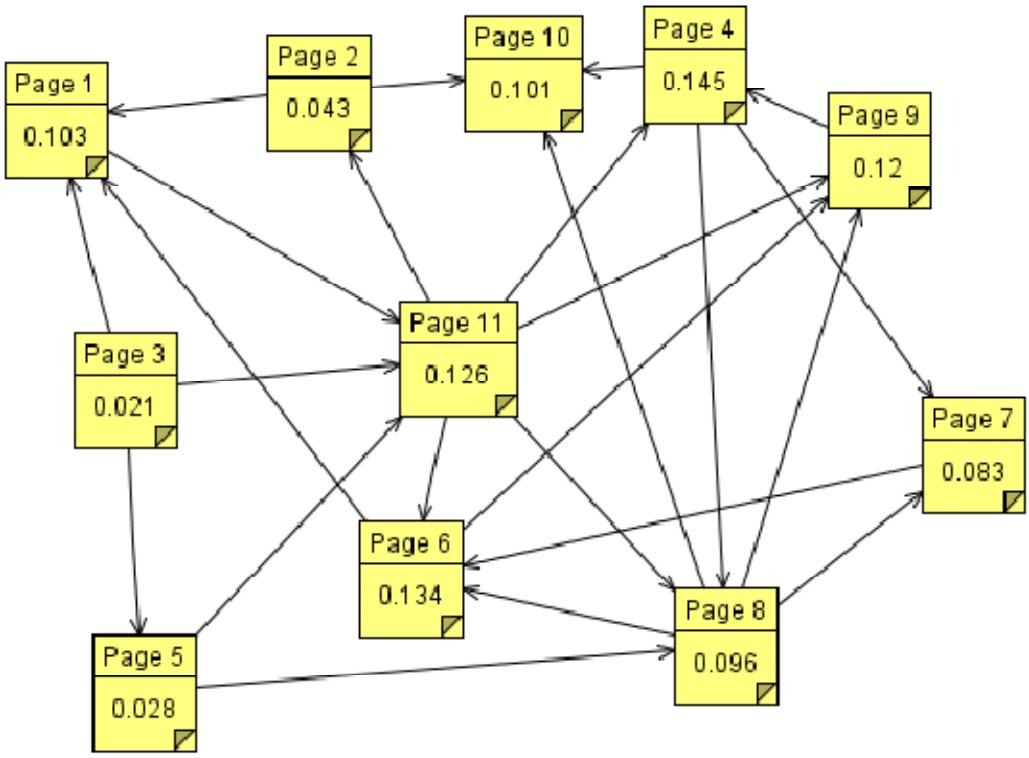

Fig. 2. Hyperlinked Structure of Web Pages with corresponding PageRanks in PageRank Sim.

After this calculation we performed text search in this collection for Boolean or a simple queries (in a form of keyword). In text search module we have 7 parameters and each parameter has its own weighted values (Table-1). For each page it will calculate simple count value of keyword for all parameters then multiply it with corresponding weighted value.

Filename	Filename count	Title	TOP 30 lines count	italic	Bold	Heading	Keyword dens...	Sum	Pagerank	Final Pagerank
15 Practical Grep Command Examples In Lin...	1	1	1	0	0	8	0.0250752256...	0.37507522567...	0.103	0.478075225677031
grep Searching for Words _ Linux Journal.htm	1	1	1	0	8	0	0.0152671755...	0.28526717557...	0.126	0.411267175572519
How to grep multiple words or strings - Linux ...	1	1	1	0	0	0	0.0127795527...	0.12277955271...	0.145	0.267779552715655
Search Text File In UNIX.htm	0	0	14	0	0	0	0.0162037037...	0.15620370370...	0.096	0.252203703703704
How can I use grep to find a word inside a fo...	1	1	1	0	0	0	0.0229147571...	0.13291475710...	0.043	0.175914757103575
linux - How to search contents of multiple pdf...	0	0	1	0	0	0	0.0181038589...	0.02810385898...	0.134	0.162103858980467
Linux Find a string in files and display just th...	0	0	0	0	0	0	0.0194444444...	0.01944444444...	0.028	0.0474444444444444
how do i search for strings from multiple files...	0	0	0	0	0	0	0.0046649703...	0.00466497031...	0.021	0.0256649703138253

Fig. 3. Search result in *MySearch* for *Grep* query

Then it will sum up all the calculated values for individual pages and will add this to their corresponding PageRank (calculated using PageRank Sim) of pages. And this will be the final Pagerank of web pages. In fig. 4 we have shown the search results of My Search for a keyword 'grep'. In fig. 4, graph shows the comparison between pagerank and final PageRank values using the above results. It shows how text search result changes the final pagerank values. It can be seen that in search results which are calculated without using text search, the web page "how to grep multiples word or string", has highest PageRank value while in final PageRank we have highest page rank value for the web page "15 practical grep command example in Linux".

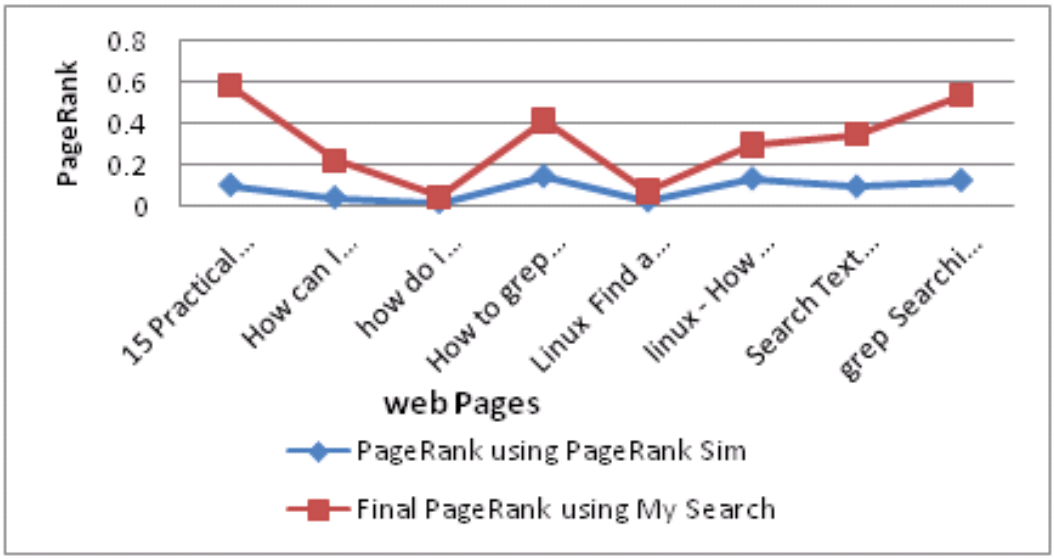

Fig. 4. comparison between pagerank and final PageRank

5. Conclusion:

Ranking module plays vital role in search engine. When a person enters a query in search engine, search engine looks on index files to get the list of web pages(which can be thousands of web pages) containing query keywords and then search engine rank them according to PageRank values of web pages. But there is a problem that all return result does not contain relevant information. Generally user access only top 10 or 20 return results of search engine for a query. In our research we use 7 parameters which can be calculated using text search to evaluate the rank of the pages. Our focus is to return only relevant results on the top. In future we can also use some other parameters to improve the ranking of the web pages.

Reference:

[1] 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.

[2] Ashutosh Kumar Singh, Ravi Kumar P, *A Comparative Study of Page Ranking Algorithms for Information Retrieval*, International Journal of Electrical and Computer Engineering 4:7 2009

[3] Neelam Duhan, A. K. Sharma, Komal Kumar Bhatia, *Page Ranking Algorithms: A Survey*, 2009 IEEE International Advance Computing Conference, 978-1-4244-1888-6/08

[4] Laxmi Choudhary and Bhawani Shankar Burdak, *Role of Ranking Algorithms for Information Retrieval.*

[5] Ashish Jain, Rajeev Sharma, Gireesh Dixit, Varsha Tomar, " Page Ranking Algorithms in Web Mining, Limitations of Existing methods and a New Method for Indexing Web Pages", 978-0-7695-4958-3/13 DOI 10. 1109/CSNT. 2013. 137/IEEE.

[6] S. Brin, and L. Page, "The Anatomy of a Large Scale Hypertextual Web Search Engine", Computer Network and ISDN Systems, Vol. 30, Issue 1-7, pp. 107-117, 1998.