Survey of Page Ranking Algorithms on Customized Searched Engine

Sharad Dixit1 ([sdixit1@umbc.edu](mailto:sdixit1@umbc.edu))

Master of Computer Science

University of Maryland, Baltimore County

**Abstract:**

**World Wide Web has become a vast resource for information, which has led to the problem of management of information on the web and satisfy the user need. It has turned out to be hard to follow the data on the web which has prompted the need of advancement of some data retrieval strategy which has led to development of some information retrieval technique which can find, extract, filter and obtain the output required by the user. Tools used presently in order to retrieve information from the WWW is a search engine like google, yahoo or Bing for information retrieval. These tools answer a huge number of queries every day and index and rank each web page hundreds of millions of webpages. User inputs his query in the search engine and then search engine fetches web pages which are desired by the user from world wide web.**

**The results provided by the search engine needs to be good quality with respect to the user and also the retrieval should be efficient, fast and accurate. In this paper, a survey of page ranking algorithm is done first and then subsequently one of the page ranking is implemented on a customized developed search engine in order to see the efficiency and relevancy of the ranking algorithm in a practical scenario.**

**Keywords- WWW; search engine; ranking algorithm; score**

1. **Introduction:**

With the exponential development of data accessible on the World Wide Web (WWW), it is getting to be hard to acquire the data from the web and further satisfy recovery needs required by the client. It has been evaluated that World Wide Web has extended more than 2000% since its development and has been multiplying in estimate in each six to ten months [2], which has made hard to follow the data on the web. As a matter of fact, we are suffocating in information however starving for learning. Consequently, with a specific end goal to fathom many advanced web searching and mining techniques have been recently developed to find, extract, filter or evaluate the relevant information as per the user need. Information retrieval tools like search engine are developed to retrieve relevant information from the World Wide Web. Some famous search engine are Google, Yahoo and Bing. User inputs his query in the search engine and the search engine retrieve results from the web. The basic architecture is shown below in Figure 1.

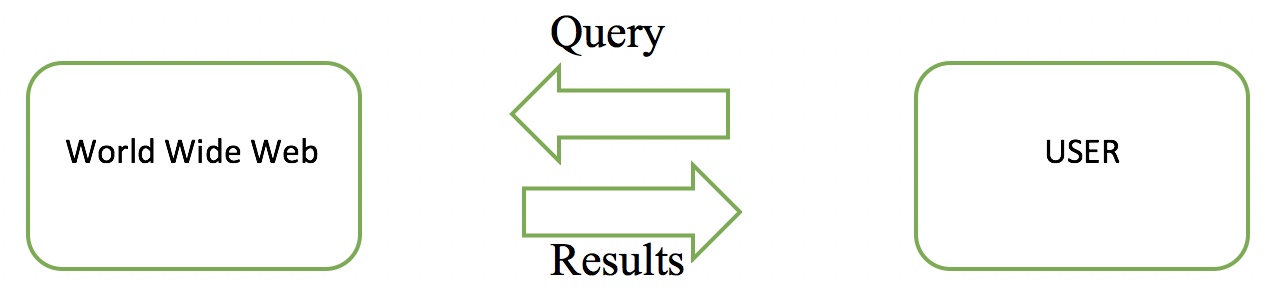


Figure 1. Overview of Search Engine

But before presenting the results to the user, some ranking algorithm (web mining) has to implemented in order to present the user with relevant results. A search engine has four core components that are crawler, indexer, page ranking algorithm and query processor. The crawler serves as one of the most important aspects of a search engine. It is also known as a spider as its function is to traverse through the entire web through the hypertext structure of the web and store information about each web page. The stored webpages are then parsed to an indexer which build the index based on the keywords present in the webpage. Index is generally maintained in an alphabetical order.

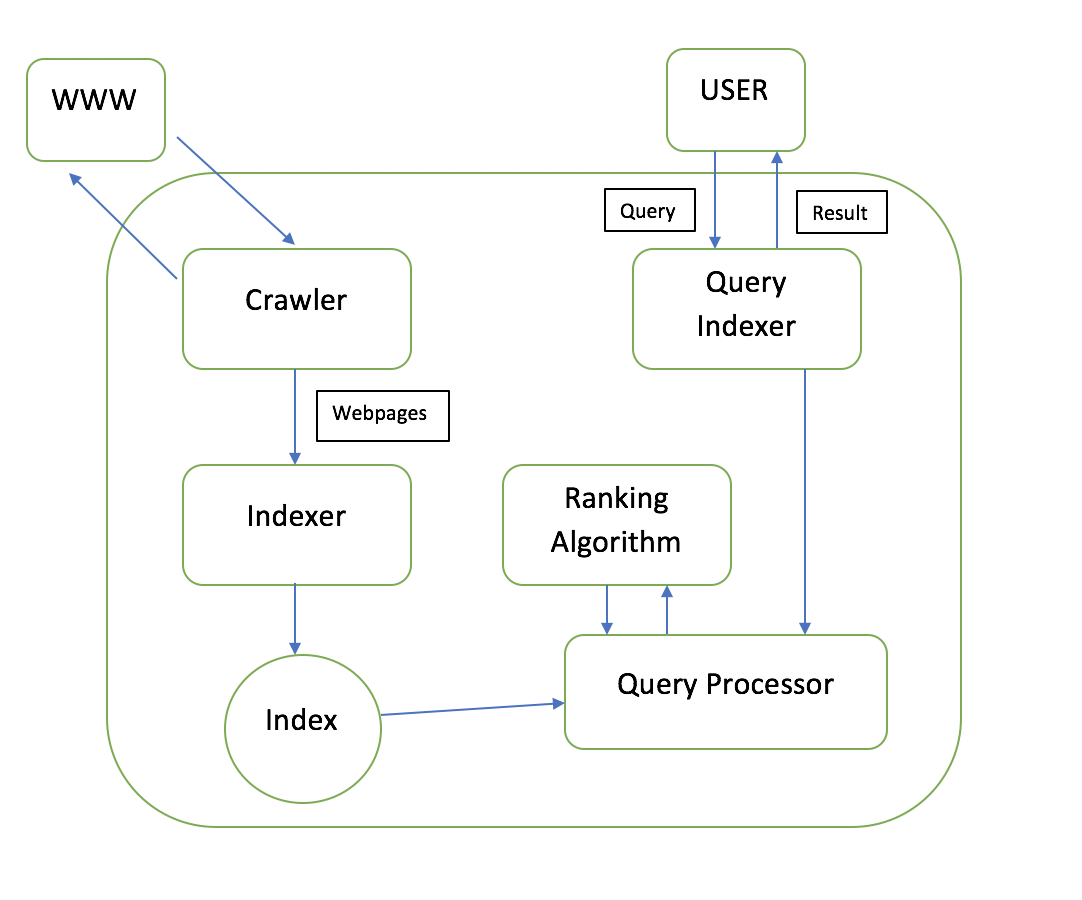


Figure 2. Architecture of a Search Engine

The search engine has a query processor component, where when a query is fired to a search engine it is preprocessed, which after matching the query keywords with the index returns the URLs of the pages to the user. Therefore, After the web pages which have keywords similar to that of the query are obtained then before representing the web pages to the user, a mechanism is needed which only shows those pages to the user which are desired by the user. That is web pages which are more relevant to the user are kept on the top and those which may not be relevant at the bottom. So, for to do this a page ranking algorithm (Web mining) is needed which can present the user with desired results. Henceforth essential outcomes are kept on the top though less vital/significant are kept at the base of the outcome list, such sort of ranking mechanism is utilized by popular web engines like Google with PageRank as a Ranking Algorithm. Therefore, the Page Ranking mechanism plays a vital role in ranking the results before presenting the results to the user. Since there have been a lot of research done on different Page Ranking Algorithm present in the market, a survey is needed which can tell the efficiency and comparison of different algorithm with respect to other algorithms.

In this paper, a survey of different page ranking algorithms is done and then subsequently one of the page ranking is implemented on a customized developed search engine in order to see the efficiency and relevancy of the ranking algorithm in a practical scenario. This paper is structured as follows: Section 2 tells about the related work done in this field. Section 3 provides a detail overview of some page ranking algorithms, Section 4 discuss the strengths and limitations of each algorithm and do an extensive comparison between each algorithm. Section 5 describe the development of customized search engine and further implementation of page ranking algorithm and present with results of customized search engine in practical scenario and further Section 6 concludes the paper with a light on future suggestions.

1. **Related Works**

In [1], the author does an extensive research on search engine and present with different component involved in developing a search engine and how each component plays a vital role in handling the functions. Further in [2], the author presents with different number of ranking algorithms currently available for the digital library and then do an extensive comparison between each algorithm on the basis of different performance as well as other search engine specific parameters. [3][4], also present with different ranking algorithms present currently and how optimized they are with respect to different parameters. They also present with different web mining techniques and how they are related to ranking algorithms. [5], [6] does an extensive research on the web mining techniques present and tell how each different technique that is web content mining, web structure mining and web usage mining differs from each other and how these techniques use different aspect like content of a web page, web link structure and usage pattern of the user respectively, in order to perform data mining that is finding useful hidden information on WWW. [7], presents with the original PageRank algorithm developed by Surgey Brin and Larry Page and how it currently also forms the base of Google. Finally [8][9] present with different ways to construct index that is different indexing techniques and how each technique has different efficiency in indexing a corpus and [10] provides a complete overview on how to develop a search engine.

1. **PAGE RANKING ALGORITHMS**

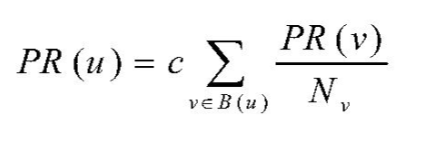
With the broad exponential development of information accessible on the World Wide Web, the quantity of clients and queries to get to the information from the web is likewise expanding. Therefore, the main task faced by the search engines is to efficiently extract the relevant information and the order those result information Subsequently, with a specific end goal to rank that retrieved data that is order the documents, various page ranking algorithms have been suggested and employed utilizing different schemes to calculate relevance and importance of a document based on different parameters. Some of the common page ranking algorithms have been discussed here as follows.

1. **PageRank Algorithm**

Surgey Brin and Larry Page [6] at Stanford University proposed an algorithm named PageRank (PR) algorithm. It has become the most commonly used ranking algorithm. It has encouraged and laid the foundation for many other ranking schemes. This ranking algorithm uses the web’s link structure to determine the importance and relevance of a web page. The algorithm states that if a page has some important incoming links to it then its outgoing links to other pages also become important. Therefore, it takes backlinks into account and propagates the ranking through links. In other words, this algorithm calculates rank of web page by considering incoming links to a web page. Thus, a page obtains a high rank if the sum of the ranks of its incoming links is high. At the point when some query is given, it combines precomputed PageRank scores with content coordinating scores to obtain an overall ranking score for each resulted web page in response to the query.

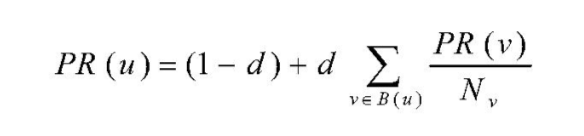
The PageRank calculation is the core of Google and considers in excess of 25 billion site pages on the World Wide Web with a specific end goal to rank the outcomes by calculating out a rank score.

A simplified version and early model of PageRank algorithm is defined as in Eq.1 from [6]:



u represents a web page, B(u) is the set of pages that point to u, PR(u) and PR(v) are rank scores of pages u and v respectively, Nv denotes the number of outgoing links of page v, c is a factor used for normalization. In PageRank the rank is equally distributed among its outgoing links. Later PageRank was modified as all the users do not follow a direct link to web structure on the World Wide Web and hence a damping factor was introduced which was usually set to 0.85.

The modified version of PageRank is defined as in Eq.2:



where “d” is a dampening factor. “d” can be thought of as the probability of users following the direct links and “(1-d)” as the page rank distribution from non- directly linked pages.

1. Strengths and Weakness of PageRank Algorithm:

One of the main strengths of PageRank algorithm is that it ranks the web pages according to the importance and relevancy of the web pages that is the more number of incoming links to a web page the more important is the page. On the other hand, there has been some weakness of this algorithm:

* The rank score of a web page is equally distributed among all its outgoing links irrespective of assigning a higher score to more relevant and important web pages.
* A page rank of a web page is mostly affected by the scores of the web pages that point to it and less by the number of web pages. For example, in Figure 3, node F gets higher score than node E, although node E gets 4 web pages pointing to it and node F gets 1 web page pointing to it.

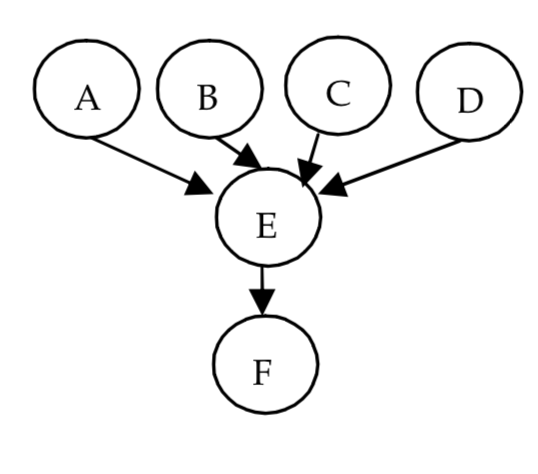


Figure 3. Example of Graph (Nodes representing Web Pages)

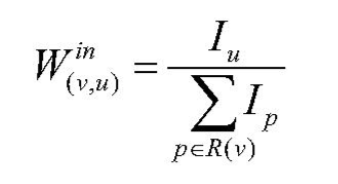
* If in a cycle, that is a web page points to some web page and further a cycle is formed, then PageRank gives a higher score to the last node of the cycle that is the node which completes and joins back the cycle.

1. **Weighted Page Rank Algorithm**

Wenpu Xing and Ali Ghorbani [2][3] proposed an extension to standard PageRank called Weighted PageRank (WPR). The assumption taken by this algorithm was that if a page is more popular, more links of other web pages point towards it or more number of links from the page go out to other web pages. Therefore, this algorithm considered both the incoming links and the outgoing links in order to calculate the popularity that is importance of a web page. This ranking algorithm does not distribute the rank equally as done in the PageRank algorithm rather it assigns larger rank to more important web page. The value assigned to each outgoing link is proportional to the respective web page popularity or importance which is seen by number of incoming and outgoing links to the web page. This importance or popularity of a web page is calculated by the number of incoming and outgoing links. The importance is assigned in terms of weight values denoted as Win(v,u) and Wout(v,u) for the incoming and outgoing links respectively.

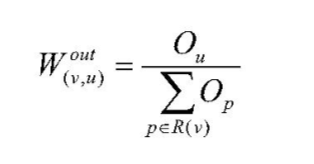
Calculation of Weighted Page Rank

Win(v,u) is calculated based on the number of incoming links of page u and the number of incoming links of all reference (outgoing linked) pages of page v. The Win(v,u) is calculated as in Eq. 3 from [2][3]:

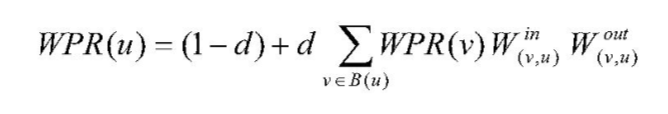


Iu and Ip represent the number of incoming links of page u and page p, respectively. R(v)denotes the reference page list of page v.

Wout(v,u) is the weight of link(v,u) calculated based on the number of outgoing links of page u and the number of outgoing links of all reference pages of page v. The Wout(v,u) is described in the Eq.4:



Ou, and Op represent the number of outgoing links of page u and page p, respectively. Therefore, the Weighted Page Rank algorithm consider the importance of web pages in terms of weights and finally modifies the original PageRank Algorithm as in Eq.5:



1. Strengths and Weakness of Weighted Page Rank Algorithm

The main strength behind this algorithm is that it considers the popularity and importance of web pages in terms of weights which was not done in original page rank algorithm and hence provide more relevancy and importance that is retrieved web pages are more relevant and important as compared to the original PageRank algorithm.

On the other hand, there has been some weakness in this algorithm:

* However, the relevancy is higher than the original PageRank algorithm, but the relevancy is still considered less with respect to the user as the method computes the scores at a single level.
* The scores are computed at the indexing time which results in unequal distribution of score.
* If in a cycle, that is a web page points to some web page and further a cycle is formed, then PageRank gives a higher score to the last node of the cycle that is the node which completes and joins back the cycle.

1. **SIMRANK: Page Rank Approach Based on Similarity Measure**

Shaojie Qiao[3][4] proposed an improved and optimized approach to rank the web pages in the query result list on the basis of the similarity measure from the vector space model named as SimRank. This algorithm utilizes the concept of social annotation [10] known a SimRank. It computes the similarity of the documents and then further apply that to the whole corpus in order to partition it. The web annotators named as SimRank associate some set of textual content with every web page so as to provide a prior knowledge regarding the web page to the web user without reading the internal contents of that page, so that the user does not have to parse the whole document again and again and with which user can navigate easily. These set of textual contents are known as annotations. The annotations are parsed contents holding the important keywords of a web page. This abstract of social annotation is known as SimRank.

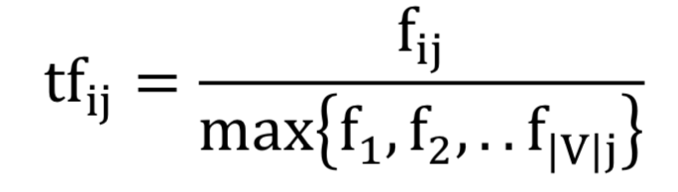
After developing of the social annotation for each web page, the pre-processing gets completed. When a user inputs query to retrieve documents, some web pages are retrieved which are both relevant and irrelevant. Then this algorithm assigns a ranking score to each retrieved document on the criteria of matching the similarity between the keywords and annotations. Simply saying, this algorithm correlates the query with the annotations of every web page and finally assigns a score on the basis of similarity between the query and annotations and hence gives importance to web pages on the basis of similarity score between the documents.

1. Similarity Measure

In order to calculate the similarity measure between the query and the annotation, we need to find the overall weight first for each term in a document, which is defined as the multiplication between the term frequency and inverse document frequency.

1. Term Frequency

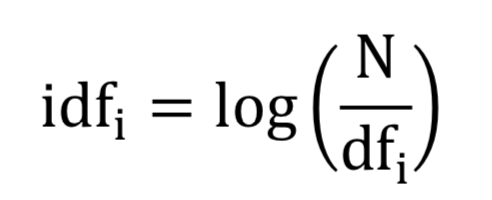
The term frequency is defined as the number of times a term occurs in a document, that is the count of a term in a document. The term frequency of a term ti in the page dj is calculated as in Eq.6:



Where fij denotes the frequency of the term in the page and |V| is the vocabulary size.

1. Inverse Document Frequency

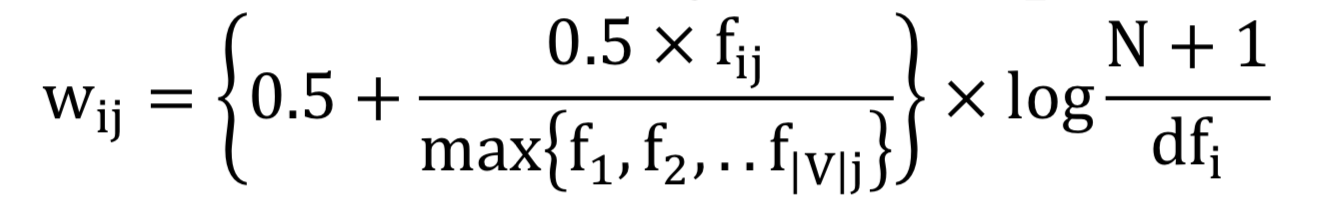
The inverse document frequency is defined as the number of documents where the term occurs. The inverse document frequency of a term ti is calculated as in Eq.7:



Where N is the total number of web pages in the web database, dfi denotes the number of web pages in which the term appears at least once.

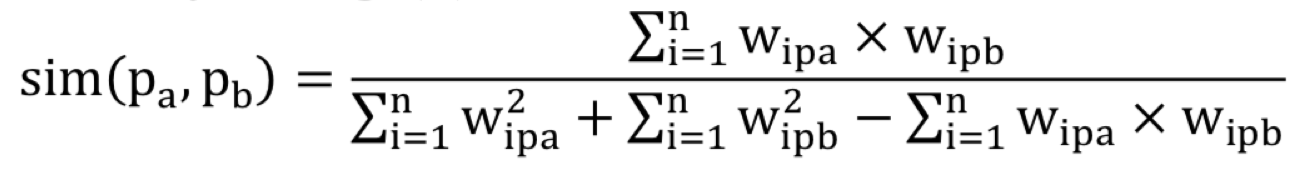
1. Overall Weight

Overall weight also known as tf\*idf weight is defined as the multiplication between term frequency(tf) and the inverse document frequency(idf). Eq.8: shows overall weight calculation.



1. Similarity Measure

Similarity between two pages paand pb is computed by using Eq.9:



1. Working of SimRank

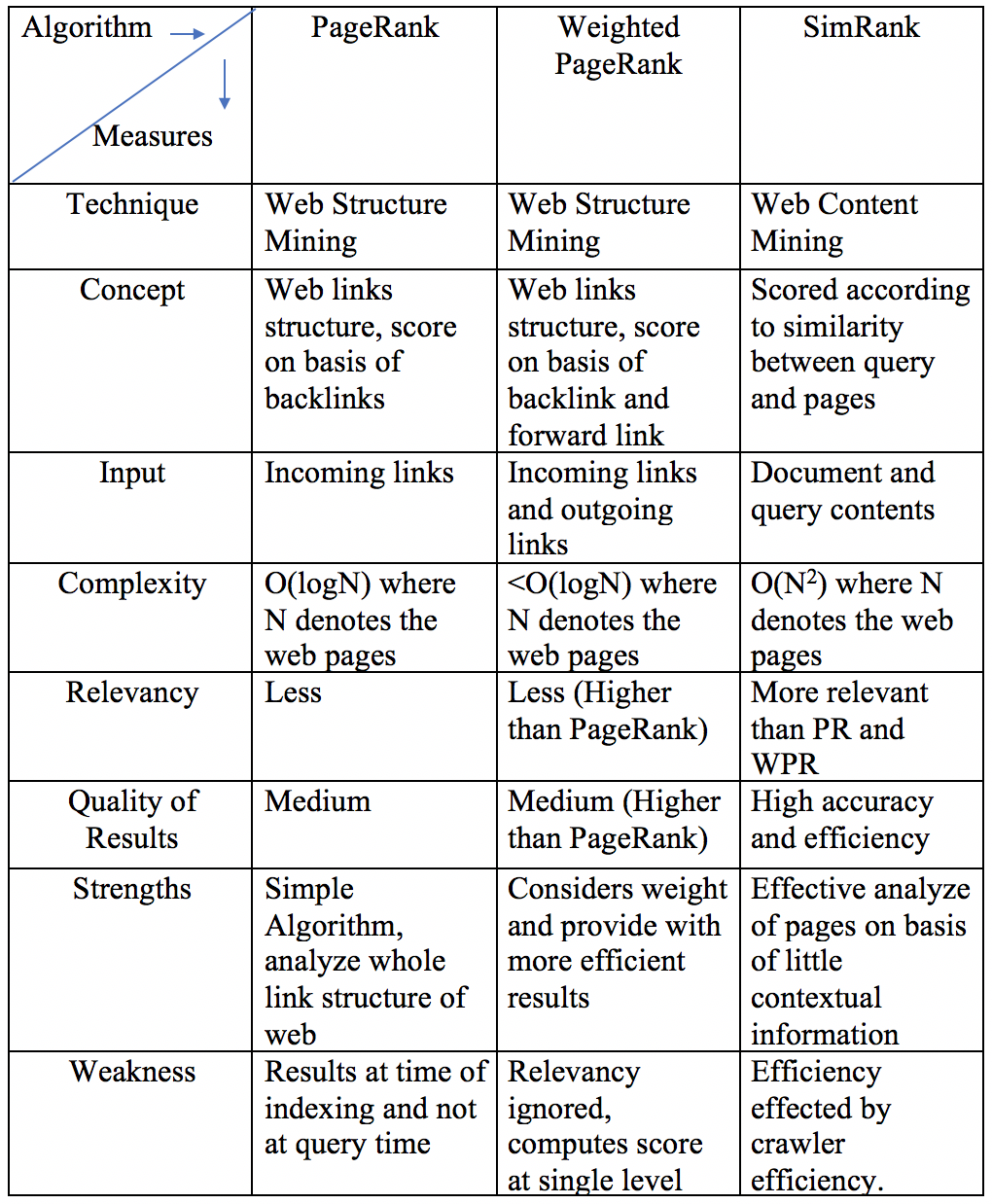
* Similarity score is calculated between the pages of the whole corpus is the first task in hand.
* Secondly, it uses the similarity score between all the documents and store it in a matrix and as the distance measure it applies k-means algorithm to form cluster with pages holding similar contents.
* Finally, it computes the similarity with respect to the query and assigns a relevance score to each web page.

1. Strengths and Weakness of SimRank:

The main advantage of this method is that it uses similarity measures to effectively cluster and score the publications. On the other hand, this algorithm has the issue that its efficiency gets affected by the capabilities of the web crawler being utilized.

1. **Comparative Study between Ranking Algorithms**

By broad examination and literature analysis, it is concluded that each ranking algorithm is unique and relevant itself. They persevere and are presently being used due to their qualities and uniqueness in ranking the documents. A detailed comparison of ranking algorithms studied is shown in Table 1. Comparison is done on the basis of some measures such as main techniques used, methodology, input parameters, relevancy, quality of results, importance and limitations.



1. **Development of Customized Search Engine with SimRank**
2. Development of Search Engine

This section describes on how to develop a search engine and then further implement SimRank for ranking of the documents and presents real time simulation results of the developed customized search engine. A search engine has four core components that are crawler, indexer, page ranking algorithm and query processor.

1. Crawler

Crawler also known as spider has the function to traverses through the entire web and store information about each webpage. This component is one of the most important part as the user needs fresh information every time he searches through the search engine and which can be achieved only if the crawler crawls the web periodically so that the index constructed further is fresh and contains new information. For building the customized engine which was basically developed in order to get the real time view of the page ranking algorithms, the corpus could be either crawled through the web or a static corpus could have been taken. In this paper, the global corpus taken was a static corpus which was a Wikipedia: Data Dump of 6GB. The global corpus can be easily shifted to World Wide Web by implementing a crawler for extraction of web pages if the search engine is to implement in practical scenario.

1. Inverted Index

The inverted index is built on the basis of tokens present in each document and is generally maintained in an alphabetical order. The stored documents are first pre-processed which involves the task of extraction of tokens from the documents with stop words like ‘the’, ‘a’ removed then stemming and finally tokens are used to create the index. The indexer module does all this task of creating an index which is basically saved as a file which can be used later when a user inputs in the search engine so that all the documents of the corpus are not needed to be parsed again and again. Therefore, involving an inverted index completely reduces the query result extraction time.

* The customized inverted index developed for the search engine is as follows:

Each token is present in the inverted index and with that the documents that token belong to are present with the frequency count of the token in each document.

* The developed inverted index is serialized as a serializable file which is later use for retrieval when the user enters the query in the search engine

**Sample Inverted Index:**

Cat(TOKEN): <FILE1 ,10(TERM FREQUENCY); FILE2, 22; FILE 3, 33; ……………; FILE N, N

1. Query Processor

This component of a search engine takes in query from the user and pre-process the query before matching it with the index developed already of the corpus. This step involves tokenization and further removal of stop words, stemming. Then the keywords of the query which contain information which can be used for query serving are matched with the index and documents are retrieved where the tokens match in the index. But now before presenting the results that is retrieved documents, some ranking mechanism needs to be applied in order to present the user with relevant results.

1. Page Ranking Algorithm: SIMRANK: Page Rank Approach Based on Similarity Measure

This algorithm computes score on the basis of similarity between the query keywords and the annotations. Simply saying, it matches the query contents with the annotation of each document. This algorithm is divided in three following steps in order to rank the documents for presenting the user with relevant results.

1. Computation of similarity between the documents of the whole corpus. The calculation of similarity between the documents is done in the following steps:
2. Term frequency calculation.
3. Inverse document frequency.
4. Overall weight calculation = term frequency \* inverse document frequency
5. Use of weights to calculate similarity between the documents.
6. Then K-means clustering algorithm is applied in order to form cluster of similar documents, this is achieved by using similarity score between documents as a distance measure between the documents.
7. Last Step involves calculation of similarity among the query and clusters and assigns a relevance score to the retrieved documents.

After all the steps above are complete the user is presented with relevant and accurate results.

1. Results of Developed Search Engine

* Inverted Index: The indexing speed depends on the data corpus, presently for Wiki data dump of 6GB, the indexing speed is near about 3.5 seconds.
* SimRank Page Rank Algorithm:
  + The results obtained using this Page Ranking algorithm are relevant with increased efficiency and accuracy in ranking of pages in the result list.
  + The algorithm effectively analyzes pages or documents with little contextual information.
  + However, its efficiency gets affected if the corpus has very less number of relevant documents with comparison to the query. Therefore, the efficiency gets affected by the capabilities of web crawler being used.
  + The complexity given by the algorithm is O(N2) where N denotes the number of papers.
* The time taken for retrieval of results is near about 2 seconds.

1. **Conclusion and Future Works**
2. Conclusion

Search engines are therefore the best tool for to retrieve information from the World Wide Web, however customization of each component of a search engine is an important task where optimization of each component is needed to get good quality results with respect to the user and also the retrieval should be efficient, fast and accurate.

Further, Ranking Algorithms play a vital role in a search engine in presenting relevant and accurate results to user and hence making navigation through the web easy for the user. The PageRank considers backlinks to a web page in order to compute the score whereas the Weighted PageRank which is an improvement over the original PageRank considers both backlinks and forward links for to compute the score. Both the algorithms hence rely on the web link structure rather than content of a web page which is done by the SimRank algorithm. Therefore, depending on the ranking algorithm relevancy to a user is obtained that is order of results may differ for each algorithm. Therefore, for getting more relevant and accurate retrieval of web pages, algorithms could employ techniques like usage pattern of a user etc.

1. Future Work

The customized search engine developed still lack in lot of areas according to present technological advancement. The following implementations could improve the search engine more and hence provide the user with more relevant and accurate results.

Implementation of more efficient Ranking algorithm in order to get more relevant and quality-based results. Implementation of Web Usage Mining with machine learning in order to provide the user with more relevant and user-oriented results. Improvement in indexing speed and query throughput by improvement in the development techniques used in developing inverted index. Incorporating crawler in the search engine and even optimizing it in order to get fast crawling and fresh results.

REFERENCES

[1] Naresh Barsagade, “Web Usage Mining And Pattern Discovery: A Survey Paper”, CSE 8331,2003.

[2] Sumita Gupta, Neelam Duhan, Poonam Bansal; A comparative study of page ranking algorithms for online digital library. International Journal of Scientific & Engineering Research, Volume 4, Issue 4, April 2013.

[3] Sergey Brin and Larry Page, “The anatomy of a Large-scale Hypertextual Web Search Engine”, In Proceedings of the Seventh International World Wide Web Conference, 1998.

[4] A.K. Sharma, N. Dhuan and G.Kumar, “A Novel Page Ranking Method based on Link-Visits of Web Pages”, Int. J. of Recent Trends in Engineering and Technology, vol. 4, no. 1, pp. 58-63, Nov 2010.

[5]  P. Thwe, “Proposed Approach For Web Page Access Prediction Using Popularity And Similarity Based Page Rank Algorithm”, international journal of scientific & technology research, vol. 2, no. 3, march 2013.

[6] L. Page, S. Brin, R. Motwani, and T. Winograd.P, “The Pagerank Citation Ranking: Bringing order to the Web. Technical report, Stanford Digital Libraries SIDL-WP- 1999-0120, 1999.

[7] N. Tyagi, S. Sharma, “Weighted Page Rank Algorithm Based on Number of Visits of Links of Web Page”, International Journal of Soft Computing and Engineering (IJSCE), vol. 2, no. 3, July 2012.the first word in a paper title, except for proper nouns and element symbols.

[8] Ding Y., “Applying Weighted PageRank to Author Citation Networks”, Journal of the American Society for Information Science and Technology, pp. 236-245.

[9] K. Wen, R. Li, J. Xia and X.Gu, “Optimizing ranking method using social annotations based on language model”, Springer Science+Business Media B.V. 2011, Jan 2012.

[10] Sumita Gupta, Neelam Duhan(Page Ranking Algorithms in Online Digital Libraries: A Survey) International Journal of Scientific & Engineering Research, Volume 5, Issue 2, April 2014.

[11] Search Engines: Information Retrieval in Practice, by Croft, Metzler and Strohman. (Journal book)

[12] V.T. Nguyen, “Using social annotation and web log to enhance search engine”, IJCSI International Journal of Computer Science Issues, vol. 6, no. 2, 2009.