IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

# HYBRID ALGORITHM FOR PAGE RANKING IN INFORMATION RETRIEVAL SYSTEMS

**Pallavi [*], Dushyant Singh**

[*] M.Tech. Scholar, Department of Computer Science, Chandravati Group of Institutuions, Bharatpur, Rajasthan, India

Dushyant Singh, Head of Computer Science Department, Chandravati group of Institutions, Bharatpur, Rajasthan, India

**DOI:** 10.5281/zenodo.154224

## ABSTRACT

Information Retrieval IR systems store a large volume of unstructured data and provide search results for a user query. The performance of the IR systems depends upon the relevancy of the search results with user query. Page ranking algorithms are used to assign rank to the retrieved results for a user query. Page ranking algorithms are mainly categories in to web structure mining and web content mining. In literature many page ranking algorithms have been proposed to improve the relevancy of search results for a user query. In this paper a new hybrid page ranking algorithm using web structure mining and web content mining has been proposed. The algorithm is implemented and tested on a test data results shows that the new proposed algorithm performs better than the existing algorithms.

**KEYWORDS**: Information Retrieval, Search Engine, Page Ranking, Web Structure Mining, Web Content Mining

## INTRODUCTION

Information Retrieval (IR) system provide searching and retrieval of data for a user query. IR system store huge amount of data which is normally of many types and unstructured in nature. Search engines such as Google are also IR systems which provide a list of URLs on which relevant information regarding a user query is expected to found. The general structure of a search engine is shown in Figure-1. Crawling, indexing and page ranking are three major components of a search engine. Crawler works as a downloader and keep downloading the web pages from WWW irrespective of the search query. Indexer module scans all the stored web pages from the local store and makes entries in the index. The query processor module processes the user query and finds a list of documents which are relevant for the user query.

The major problem in the search engine the huge amount of data stored. For a user query the search engine usually retrieves many lacks of web pages which are relevant to the user query. The page ranking module solves this problem. Page ranking module uses its special algorithms to rank web pages according to their relevancy to the user query. Better is the page ranking module, better will be the results of a search engine as the algorithm assign highest rank to those pages which are most relevant for the query. Web structure mining, web content mining and web usage mining are the three major categories of page ranking algorithms [3]. The web content mining algorithm calculates the relevancy of the web pages by scanning the content of the web pages i.e. the data stored in web pages. Whereas the web structure mining algorithm calculate the importance of a web page by analyzing the inlink and outlinks of the web pages i.e. from which pages there is an hyperlink to that web page(inlinks) and to which pages there is a hyperlink from the web page(Outlink). Web usage mining algorithm works on the web usage history of the user. It creates a history of the web usage for the user and by analyzing that history it calculates the relevancy of a web page for the user query.

ISSN: 2277-9655
Impact Factor: 4.116

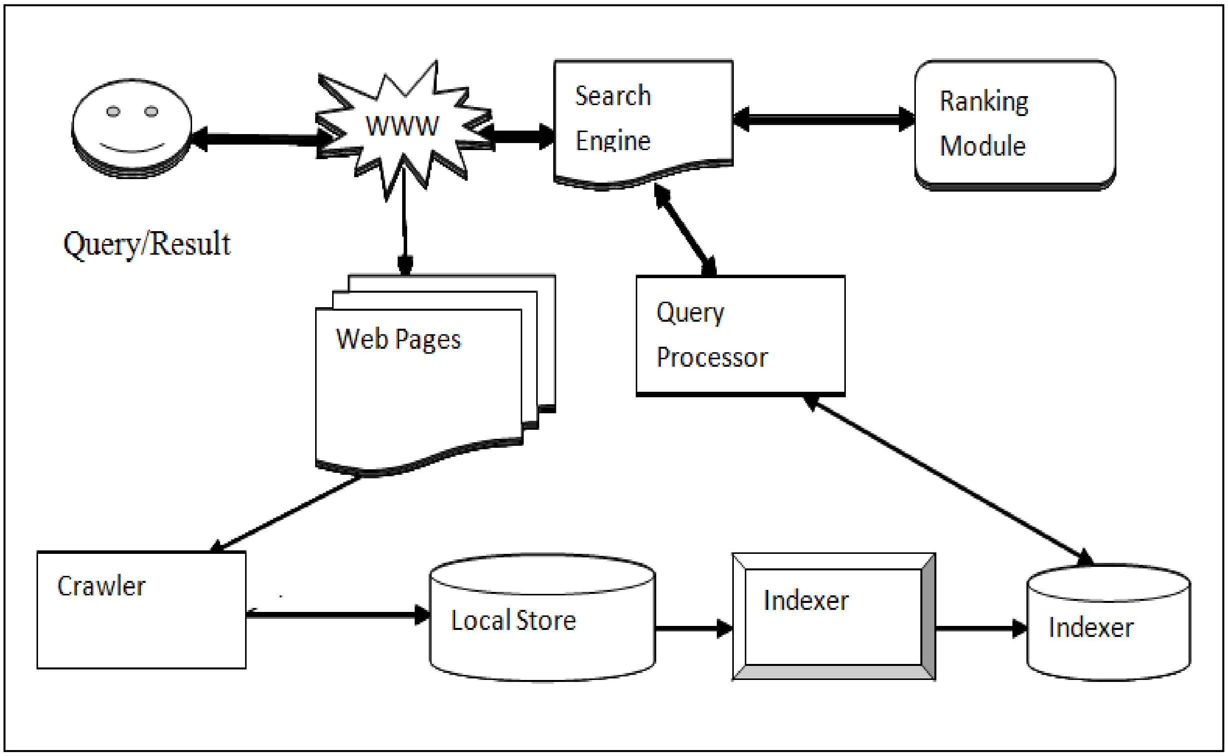


*Figure-1 General Structure of a Search Engine*

Many page ranking algorithms are proposed by various authors. Er.Tanveer Singh and Dr. Raman Maini [1] proposed a new algorithm for page ranking. In this paper the author proposed that after retrieving the web pages for a user query a further work is done to improve the quality of search results. Author uses page keywords and count these words to improve the relevancy of the search results. P.Sudhakar, G.Poonkuzhali and R.Kishore Kumar [2] proposed a content based page ranking algorithm for search engine. The algorithm create a dictionary for the user query by the keywords and content extracted from each link. Then author calculate the weight of keywords and content against the dictionary. Then algorithm rank each link on the basis of the weight obtained. Neelam Tyagi and Simple Sharma [3] compare different web structure mining algorithms for page ranking. Author first illustrate the web structure mining algorithms such as Page Rank algorithm and Weighted page rank algorithm and then compare various algorithms on some parameters such as methodology, key in parameter, importance and limitation etc. Author compare hit based algorithm, page rank algorithm and weighted page rank algorithm on these parameters and concluded that different algorithm performs better for different parameters. After analyzing the literature it has been concluded that the page ranking algorithms plays an important role in searching information in a search engine. In this paper a new page ranking algorithm is proposed for search engine.

## MATERIALS AND METHODS

In this work a Hybrid Page Ranking Algorithm has been proposed. This algorithm combines two techniques of page ranking which are Web Content Mining and Web Structure Mining. The architecture of Information Retrieval System (Search Engine) that uses proposed Hybrid Page Ranking Algorithm has been shown in Figure-2.

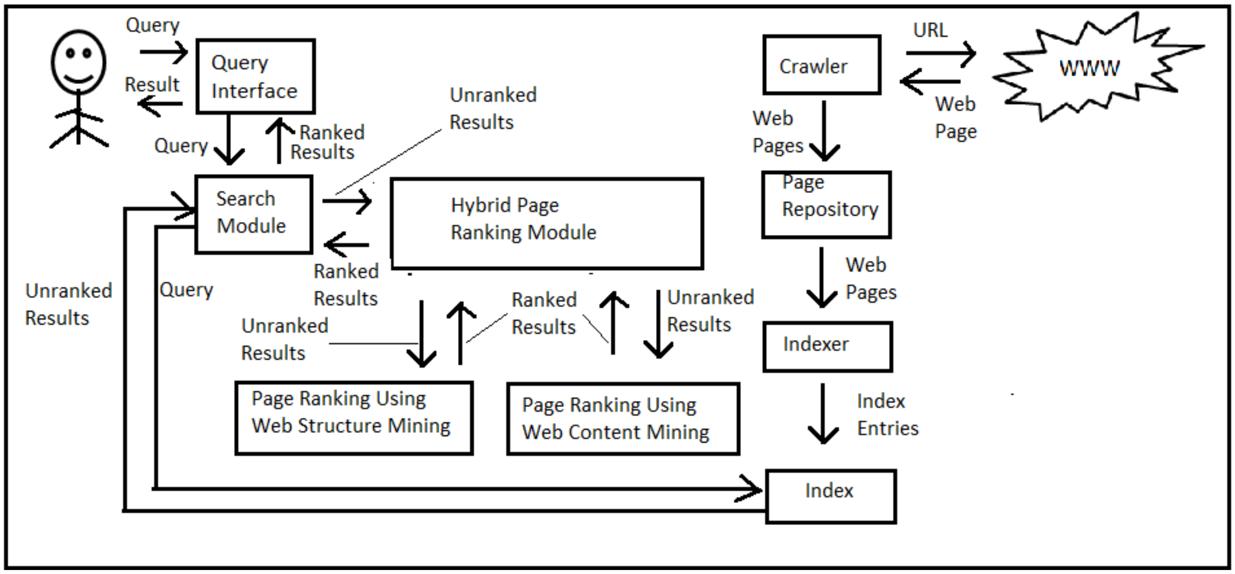


*Figure-2 Proposed Architecture of IR System Using Proposed Hybrid Page Ranking Module*

The major components of the proposed architecture are as follows:

1. Query interface
2. Search Module
3. Crawler
4. Page Repository
5. Indexer
6. Index
7. Hybrid Page Ranking Module
8. Page Ranking Using Web Structure Mining
9. Page Ranking Using Web Content Mining

The working of these modules is as follows:

### Query interface

This module will receive a user query from the user. It will parse the query and find out all the query words and then forward this query to the Search Module. Then the search module will find all the related web pages, find rank of these results according to their relevancy to the user query and then return these ranked results to the query interface.

### Search Module

This module will receive the query from the Query Interface and find all the related documents to this query from the Index. Then it will provide these unranked results to the Hybrid Page Ranking Module. The Hybrid Page Ranking Module will return the ranked results to the Search Module. Then the Search Module provides these ranked results to the Query Interface.

### Crawler

This module crawl the World Wide Web and download the web pages from the World Wide Web. This module stores the downloaded web pages in a page repository. This module works in the background independent to the user query.

### Page Repository

All the downloaded web pages have been stored in the page repository by the crawler. This page repository is used by the indexer module.

### Indexer

This module takes downloaded web pages from the page repository. This module parses all the web pages one by one, extract all the tokens from the web pages and then create an index. It makes entry in the index for all the web pages stored in the page repository.

### Index

Index contains entry for all the web pages in the page repository. It makes the searching task faster. The search module send the query to the Index and then Index will return all the web pages which are relevant to the user query. It proved an unranked list of web pages which are relevant to the user query.

### Hybrid Page Ranking Module

This is the module which performs the page ranking. It ranks all the web pages according to their relevance to the user query. As the size of web is very large so Index module usually return a large list of unranked web pages which are relevant to the user query. The quality of search engine mainly depends upon the performance of page ranking module. This module uses two techniques of page ranking which are Web Structure Mining and Web Content Mining. It first finds the rank of the web pages using Web Structure Mining and Web Content Mining separately. Then it combines the two ranks to find the final rank. So this proposed Hybrid Page Ranking will take the advantage of both the techniques and hence provide ranks to the web pages according to their relevancy to the user query.

### Page Ranking Using Web Structure Mining

This module will calculate the page rank of the given set of pages using web structure mining. This module uses web structure of the web pages to determine the importance of the web page. It uses how many inlinks are there on the web page that is how many other web pages having a hyperlink to this page. It also calculate outlinks of the web page i.e. to how many other web pages this page has a hyperlink. It then uses some formulas to calculate the page rank of the web page.

### Page Ranking Using Web Content Mining

This module calculates the page rank of the web pages by using content of the web pages. These algorithms does not rely on the inlink and outlink structure of the web pages but mainly calculate the page ranks by parsing the web page and extracting the words from it. There are many algorithms in the literature in web content mining. These algorithms uses content such as frequency of a word in the web page, ontology repositories, thesaurus and word net etc to find the ranks of the web page.

### Proposed Algorithm

This work mainly defines the algorithm used in calculation of page rank of web pages. It proposes following three algorithms to calculate the page rank of web pages.

Algorithm-1 : Calculation of page rank using Web Structure Mining

Algorithm-2 : Calculation of page rank using Web Content Mining

Algorithm-2 : Calculation of page rank using Hybrid of Web Content Mining and Web Structure Mining

The steps used in these algorithms are as follows:

**Algorithm-1 : Calculation of page rank using Web Structure Mining**

This Algorithm calculated Page Rank and then Weighted Page Rank of the web pages.

**1. Calculation of Page Rank [3]**

This algorithm first calculate the Page Rank of the web pages using the following formula

$$PR(u) = (1-d) + d \sum_{v \in B(u)} \frac{PR(v)}{N_v}$$

Where d is the damping factor and usually its value is 0.85.

PR(u) is the Page Rank of Page u and $N_v$ denotes the number of outgoing links of page v.

**2. Calculation of Weighted Page Rank[3]**

The formula for calculating weighted page rank is as follows

$$PR(u) = (1-d) + d \sum_{v \in B(u)} PR(v) W^{in}_{(v,u)} W^{out}_{(v,u)}$$

……….(1)

Where $W^{in}$ and $W^{out}$ are calculated as follows:

$$W^{in}_{(v,u)} = \frac{I_u}{\sum_{p \in R(v)} I_p}$$

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

And

$$W^{out}_{(v,u)} = \frac{O_u}{\sum_{p \in R(v)} O_p}$$

where Ou and Op represent the number of outlinks of page u and page p, respectively. R(v) denotes the reference page list of page v.

So equation (1) is used to calculate the weighted page rank of the web pages. In the Figure – 2 the module Page Ranking Using Web Structure Mining uses the formula in equation (1) to calculate the page rank of the web pages.

**Algorithm-2 : Calculation of page rank using Web Content Mining**

Assumption : This algorithm assume that the Index stored is an Inverted Index. This index store a list of tokens and a list for each token that store the Document in which this token occur with its frequency in that document.

**Algorithm : Page_Rank_by_Web_Content_mining()**

```
{
        Step 1 : read the query from the user.
        Step 2: Find the list of documents in which that query word occur.
        Step 3: Find the Frequency of the occurrence of the query words in the document s.
        Step 4: Sort the documents by the frequency in ascending order.
        Step 5: The rank of the document in this sorted list will give us the rank of the document according to web content mining.
}
```

**Algorithm-3 : Calculation of page rank using Hybrid of Web Content Mining and Web Structure Mining**

This algorithm will calculate the page rank of the web pages using a Hybrid Technique of Web Structure Mining and Web Content Mining. It first calculates the rank of the web pages using these two techniques and then calculates the sum of ranks of the web pages in these two techniques. It then calculates the average of both the ranks. Then it sort the documents on the ascending order of the average ranks. The rank of a document in this descending list will give us the final rank of the web page. The steps of the algorithm are as follows:

**Algorithm : Page_Rank_by_Hybrid_Page_ranking()**

```
{
        Step 1 : Find the rank of the web pages using Web Structure Mining and Web Content Mining
        Step 2: Calculate the sum of both the ranks of a web page.
        Step 3: Divide the sum by 2 to find the average_weight of the web page.
        Step 4: Sort the documents in the descending order of the average_weight.
        Step 5: The rank of the document in this sorted list will give us the final rank of the document.
}
```

## RESULTS AND DISCUSSION

The proposed work has been implemented in Java using JDK1.7 and NetBeans IDE 8.0.2. The work has been implemented on a test data of 10 web pages. These pages has been created and stored in a page repository i.e. folder and then the proposed hybrid algorithm and existing weighted page rank algorithm has been implemented. The value of precision and recall has been calculated for the existing weighted page rank algorithm and proposed hybrid page rank algorithm has been proposed. The details of the results are as follows:

### Precision

In the field of information retrieval, precision is the fraction of retrieved documents that are relevant to the query. It is calculated as follows:

Precision = (|{Relevant Documents}∩{Retrieved Documents}|) / |{Retrieved Documents}|

### Recall

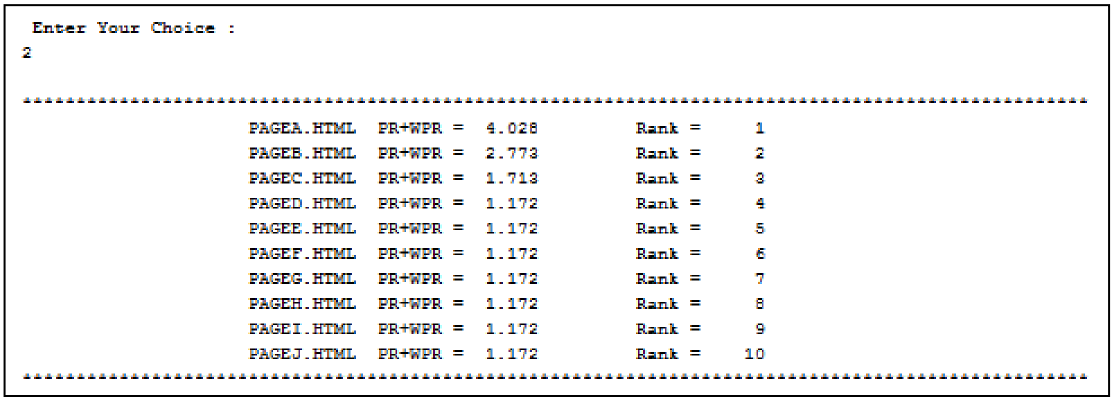
Recall in information retrieval is the fraction of the documents that are relevant to the query that are successfully retrieved.

Recall = (|{Relevant Documents}∩{Retrieved Documents}|) / |{ Relevant Documents}|

Calculation of precision and recall for existing weighted page rank algorithm is as follows

Precision
Retrieved Documents = 10
Relevant Documents = 7
Precision = 7/10 = 0.70

Recall
Retrieved Documents = 10
Relevant Documents = 7
Precision = 7/7 = 1.0
Figure-3 shows a snapshot of retrieved 10 documents for the for existing weighted page rank algorithm



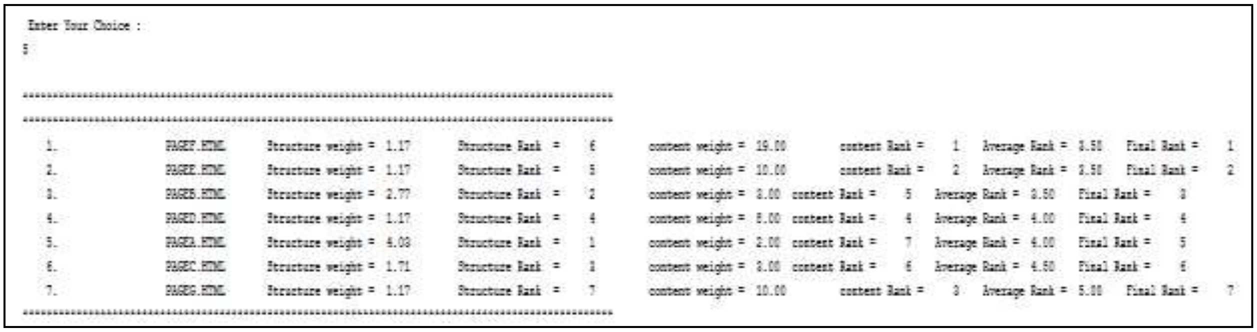
```
 Enter Your Choice :
2

*********************************************************************************************

                  PAGEA.HTML  PR+WPR =  4.028        Rank =    1
                  PAGEB.HTML  PR+WPR =  2.773        Rank =    2
                  PAGEC.HTML  PR+WPR =  1.713        Rank =    3
                  PAGED.HTML  PR+WPR =  1.172        Rank =    4
                  PAGEE.HTML  PR+WPR =  1.172        Rank =    5
                  PAGEF.HTML  PR+WPR =  1.172        Rank =    6
                  PAGEG.HTML  PR+WPR =  1.172        Rank =    7
                  PAGEH.HTML  PR+WPR =  1.172        Rank =    8
                  PAGEI.HTML  PR+WPR =  1.172        Rank =    9
                  PAGEJ.HTML  PR+WPR =  1.172        Rank =   10
*********************************************************************************************
```

*Figure-3 Snapshot of retrieved 10 documents for the for existing weighted page rank algorithm*

Calculation of precision and recall for proposed hybrid algorithm is as follows

Precision
Retrieved Documents = 7
Relevant Documents = 7
Precision = 7/7 = 1.0

Recall
Retrieved Documents = 10
Relevant Documents = 7
Precision = 7/7 = 1.0
Figure-4 shows a snapshot of retrieved 7 documents for proposed hybrid algorithm



```
 Enter Your Choice :
5

*******************************************************************************************
*******************************************************************************************
   1.        PAGEF.HTML   Structure weight =  1.17   Structure Rank  =   6   content weight =  19.00       content Rank =   1   Average Rank =  3.50   Final Rank =   1
   2.        PAGEE.HTML   Structure weight =  1.17   Structure Rank  =   5   content weight =  10.00       content Rank =   2   Average Rank =  3.50   Final Rank =   2
   3.        PAGEB.HTML   Structure weight =  2.77   Structure Rank  =   2   content weight =  3.00  content Rank =   5   Average Rank =  3.50   Final Rank =   3
   4.        PAGED.HTML   Structure weight =  1.17   Structure Rank  =   4   content weight =  8.00  content Rank =   4   Average Rank =  4.00   Final Rank =   4
   5.        PAGEA.HTML   Structure weight =  4.03   Structure Rank  =   1   content weight =  2.00  content Rank =   7   Average Rank =  4.00   Final Rank =   5
   6.        PAGEC.HTML   Structure weight =  1.71   Structure Rank  =   3   content weight =  3.00  content Rank =   6   Average Rank =  4.50   Final Rank =   6
   7.        PAGEG.HTML   Structure weight =  1.17   Structure Rank  =   7   content weight =  10.00       content Rank =   3   Average Rank =  5.00   Final Rank =   7
*******************************************************************************************
```

*Figure-4 Snapshot of retrieved 10 documents for the for proposed hybrid algorithm*

***Table 1. Comparison of weighted page rank algorithm and proposed hybrid algorithm***

| | **Parameter** | |
|---|---|---|
| **Algorithm** | **Precision** | **Recall** |
| Existing Web Structure Mining Algorithm | 70% | 100% |
| Proposed Hybrid Page Rank Algorithm | 100% | 100% |

## CONCLUSION

It has been concluded from the values of precision and recall that the value of recall is same for both the existing weighted page ranking algorithm and proposed hybrid algorithm for a sample data set of 10 web pages. But the value of precision is 0.70 for existing algorithm and 1.0 for proposed algorithm. So the proposed algorithm works better than the existing algorithm. In future the proposed algorithm can be checked for a real world data set of web pages downloaded from the WWW. Further the proposed algorithm can be compared to other page ranking algorithms such as hit based algorithm, web usage mining algorithms etc.

## ACKNOWLEDGEMENTS

I am thankful to Mr. Dushyant Singh for his continuous suggestions and guidance to complete this work without which this work was not possible. I want to say thank to all my teachers and my friends for their help and motivation to do this work.

## REFERENCES

[1] Er. Tanveer Singh et.el. ," Improving Page Rank Using Semantic Relevance", in IJAIR Vol. 2 Issue 3,2013

[2] P.Sudhakar, G.Poonkuzhali and R.Kishore Kumar, " Content Based Ranking for Search Engines", in proceeding of the International Multi Conference of Engineering and Computer Scientiests 2012, VOL I, IMECS 2012, Hong Kong

[3] Neelam Tyagi, Simple Sharma, "Comparative study of various Page Ranking Algorithms in Web Structure Mining (WSM)", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-1, Issue-1, June 2012

[4] Laxmi Choudhary and Bhawani Shankar Burdak ," Role of Ranking Algorithms for Information Retrieval", International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.4, July 2012

[5] Sumita Gupta, Neelam Duhan, Poonam Bansal, " A Comparative Study of Page Ranking Algorithms for Online Digital Libraries", International Journal of Scientific & Engineering Research, Volume 4, Issue 4, April 2013