

Hybrid Two Phase Page Ranking Algorithm for Ordering the Web Pages Based on Content and Usage Mining

M. Usha , N.Nagadeepa

Abstract— Many existing page ranking algorithms are used in web mining to display the result in search engine result page. But these existing algorithms are either based on the inlinks and outlinks of the page or content of the page. It never consider the user interest on the page to calculate the page rank of that particular page. This leads to a need of web page ranking algorithm concerning content and usage of the pages. Proposed algorithm focuses on title, Meta, H1 and paragraph tags to find the similarity of the page. Tag Analyzer Algorithm is used to analyze these tags' content. Besides, it considers how long the user stays in that page to compute user interest score. TPPR technique computes the score in two phases. Based on the output of TPPR algorithm, the URLs are sequenced and displayed to the user. Event Explore technique detects whether the user is idle or active on the page. The proposed algorithm produced better performance and displays the most relevant web pages in the top of the result. From the results, "Two Phase Page Ranking" (TPPR) algorithm is better than PR algorithm of taken data set. TPPR algorithm can be used in web mining to improve the ranking system of search engine.

Index Terms— Content Mining , Page Rank, TPPR, Usage Mining

I. INTRODUCTION

World Wide Web is a major source to retrieve information. It has very large amount of data. For the user, it is a difficult task to find a particular page or a set of pages from this massive amount of data. It is highly impossible to remember the URLs of all these web pages. To solve this difficulty, search engine acts as an intermediary between the users and WWW. The search engines are able to extract the information from the WWW for the users' given query. They are used to collect the web pages and display them to the user. The order of the URLs depends upon the rank given to the pages. To perform this task, search engines use different page rank algorithms. Some ranking algorithms use content mining and some use web structure mining. But all the existing algorithms have some limitations. This paper presents a well-organized web page ranking algorithm. It is based on content and usage information of the pages to

improve the sequence of the URLs. First, we fetch the relevant web pages from the search engine result page for the given user query. Extracted pages are stored in our database. After getting the relevant web pages from our database, our proposed algorithm is applied on the pages to order the URLs. Finally ordered URLs are displayed to the user.

II. EXISTING WORK

Existing work uses the advantage of full word matching against Dictionary. User request is processed for search engine to obtain the results. Search results are extracted and sent for pre-processing. After pre-process, Dictionary is built for user query with synonyms. Every result of the keywords and content words are compared against dictionary by full word matching. If a match is found then a point is awarded to each words based on their position using weighted technique. Finally all matched keywords and content words are summarized and normalized. [1]

Disadvantages:

- It is a static algorithm.
- It considers only relevancy of the document.
- Consumable data is enormous and the algorithm is not fast enough.
- It should include the pages' usage information also.

III. PROPOSED WORK

User query is processed in search engine to fetch the results. Top 15 URLs are fetched from search engine result page. Our proposed algorithm computes the rank of a web page in two phases. In the first phase, score will be calculated based on the content relevancy and in the second phase rank will be given based on the user access time. By adding these two scores the total rank of the web page can be obtained. At last, the normalized value of each result page is sorted in descending order to get the most relevant page on the top most place. Similarity rank determines the relevance of a page with respect to query terms by counting the number of occurrences of the query terms within the web document. It gives weight based on the locality of the keyword.

Manuscript published on 28 February 2018.

*Correspondence Author(s)

M.Usha, Research Scholar, Bharathiar University, Coimbatore, Tamilnadu, India., (e-mail: ushachandran20@gmail.com).

Dr. N.Nagadeepa, Principal, Karur Velalar College of Arts and Science for Women, Karur, Tamilnadu, India., (e-mail: nagadeepa1012@gmail.com).

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

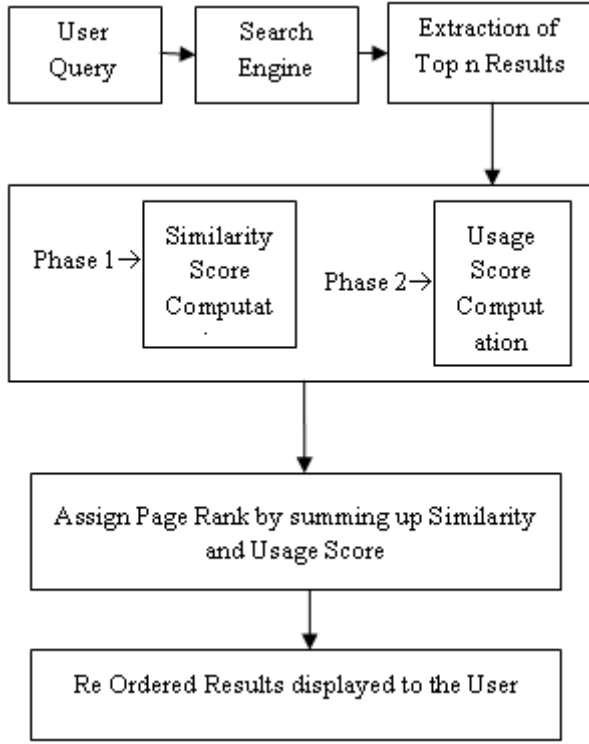


Fig 1. Architecture design

Proposed Algorithm

The HTML source of the web page is downloaded and parse it as DOM tree. DOM is an interface which allows scripts and programs to dynamically access and handles all the elements such as content, structure and style of web pages. We navigate through the DOM tree to identify title, Meta, Heading and paragraph tags. Title tag is an HTML component to identify the title of a web document. The Meta tag gives the basic information about the HTML document. It is used to state page description, keywords of the page, author of the document, last modified date etc. The H1tag will usually consist the title of a web document. It may also contain other underlined text of the page. Since these tags are good source of information about the page, they are considered in this method. They contain texts which are relevant to the content of the web page they describe. [9]

Algorithm 1: Tag Analyzer Algorithm

Step 1: Extracting Content from Title and Meta Tag

1. Input the raw HTML page P to be processed
2. Build the tree
3. Navigate the nth hierarchy nodes, T is the total number of the nodes in the n hierarchy
4. $T \leftarrow$ Number of Nodes in x
5. tt - <title> tag
6. mt - <meta> tag
7. for i \leftarrow 1 to M
8. if(Node[i].Text = tt)
9. $X = \{x \downarrow(i) \mid i \in [1, n], qk\}$
 // content mined from title tag
 // n is the number of words in the title tag
10. $sc1 \leftarrow nq / n$
 // nq is the number of query words in the title tag

11. end if
12. if(Node[i].Text = mt)
13. $Y = \{y \downarrow(i) \mid i \in [1, m]\}$
 // content mined from Meta tag
 // m is the number of words in the Meta tag
14. $sc2 \leftarrow mq / m$
 // mq is the number of query words in the Meta tag
15. end if
16. end for

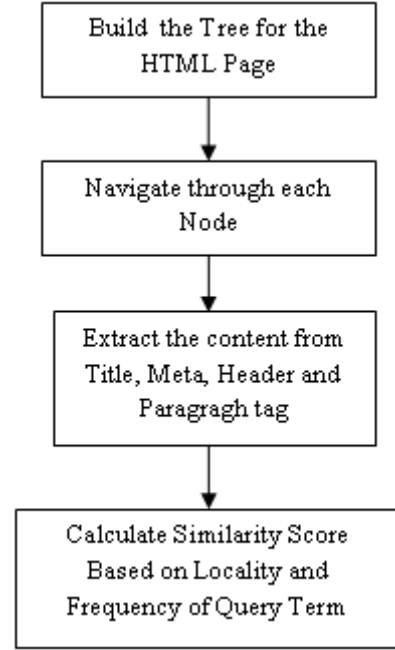


Fig 2. Workflow of Tag Analyzer Algorithm

Step 2: Frequency in Heading Tags

Headlines and important segments are usually more highlighted in the body of the web page. [8] The proposed algorithm considers Query Keyword qki that appears in header tags (H1, H2, H3... H6) is more essential than other tags. It first navigates through the entire page and fetches the content of all header tags. Then it compares the texts to search whether the Query Keyword qki appears within heading tags.

$$F(qk_i) = \sum_{j=1}^6 (s_i f_i) \quad (1)$$

where fi is the frequency of appearance of qki in header i and Si is the score of header i. Similarly to the scores are fixed to values (6, 5, 4, 3, 2, 1) respectively. The score F is then normalized to the scale of [0, 1] by the following formula:

$$S_3 = \frac{F(qk_i) - F_{min}}{F_{max} - F_{min}} \quad (2)$$

where $F_{min} = \min(F(qk1), F(qk2) \dots F(qkp))$ and $F_{max} = \max(F(qk1), F(qk2) \dots F(qkp))$ for all values.

Step 3 : PCExtractor

The tags removed include <head>, <script>, <style>, , <i> and so on. The algorithm looks at each line and creates the block using Line-block concept. Then it computes features for all blocks to decide whether they are content or not. TTR and ATTR formulas are used. If keyword density is greater than the threshold then the algorithm adds it in the output block. After calculating the content features, the system decides whether the block is content or not. This will be done based on the features values. The proposed system uses threshold methods to categorize the main content and non-content. Finally the results are analyzed. The threshold method uses standard derivation method. Threshold methods use three thresholds for TTR, ATTR and TKD. If $TTR > TTR's \text{ threshold}$ and $ATTR < ATTR's \text{ threshold}$ and $TKD \geq TKD's \text{ threshold}$ then the block is main content [7]. Otherwise, the block is noise block. After extracting the more accurately main contents, page score on the basis of paragraph content can be computed.

1. Input: Raw HTML Source code
2. $FP \leftarrow \text{remove_useless_tags and empty lines}$
3. $DB \leftarrow \text{get_blocks}(FP)$
4. For each block in DB do
5. $|TS|_{FB} \leftarrow \text{obtain_feature}(\text{block})$
6. If $|TS|_{FB} \geq \beta$ then
7. $\text{exContent.append}(\text{block.text})$
8. End for
9. $nt \leftarrow \text{number of words in exContent}$
10. $nqt \leftarrow \text{number of query words in exContent}$
11. $s4 \leftarrow nqt / nt$

Algorithm 2 : Event Explore Algorithm

1. Start Timer
2. Initialize IdleState=false;
3. Initialize idleTimer=null;
4. Idletimeout=3000; // 5 mins;
5. If (page.event = true)
6. Idletimer.reset;
7. Else
8. Idletimer=idletimer+1
9. If idletimer>=idletimeout
10. Timer.Reset = true
11. Else
12. Page.AcsTime = Timer.value

Event Explore technique is used to record the user access time of a web page. When a web page is loaded into the user's browser, timer will be triggered. Every second the timer will invoke this function. This function checks whether the user is in idle state or in active state. This verification is done by binding mouse events and keyboard events. If the user is idle continuously for 5 minutes (ie), if there is no event occurs on a page, then the timer will be reset. Otherwise the user access time will be computed using the timer value.

Algorithm 3 : Two Phase Page Ranking (TPPR)

1. Phase 1 : Compute Similarity Score

Content Weight is based on how many terms in different web page fields match with the Query keywords. The content weight is calculated differently for different types of pages to give more weightage to page characteristics.

13. Initialize SimRank=0
14. 2. For $k = 0$ to n do
15. 3. If page Pgk contains Kwd
16. 4. Insert Pgk in lop
17. 5. End if
18. 6. End for
19. 7. For $j = 0$ to lop do
20. 8. Find locality of Kwd for each page $Pg j \in lop$
21. 9. Calculate $\text{SimRank} = 0.2 * S3 + 0.3 * S1 + 0.4 * S2 + 0.1 * S4$
22. 10. End for

Every page containing the query term is added to the list of the pages [lop]. Each page from the obtained list of pages has been examined for finding the location of the keyword. The keyword occurs in Meta tag gets more weight than the keyword occurs in title tag. The keyword occurs in title tag gets more weight than the keyword occurs in heading tag. The keyword occurs in heading tag gets more weight than the keyword occurs in paragraph tag. Finally all the weights are summed up. The final score [SimRank] is the score given to the page based on the content.

Phase 2 : Compute Usage Score

1. Initialize AcsTime=0 for each page Pgk
2. α - Minimum Threshold Value
3. γ - Maximum Threshold Value
4. For $k=0$ to lop do
5. $\text{AcsTime} = \text{AcsTime} + \text{AcsTime}(Pgk)$
6. If $\text{AcsTime} > \gamma$
7. $\text{AcsTime} = \text{MaxTh}$
8. Else if $\text{AcsTime} < \alpha$
9. $\text{AcsTime} = 0$
10. End for

In phase 2, score is calculated based on the user access time of a web page. If the access time is greater than the threshold value, then it will be assigned to maximum value otherwise it will be assigned to 0. Final page rank value can be calculated by using,

$$PRV = 0.6 * \text{SimRank} + 0.4 * \text{AcsTime} \quad (3)$$

In phase 1, Content based rank is computed and in phase 2, rank is computed based on the user access time.

At last, both ranks are added together to get the total rank of a web page. Here 60% of content score and 40% of user access time are considered to get the final score.

IV. RESULTS AND ANALYSIS

A database has been created to assess the proposed TPPR algorithm. Google search engines algorithm has been chosen for comparing the proposed algorithm because same database can be used for it. Experiment is conducted with user query ("Computer") against specific search-engine. Top 7 web pages from that search-engine are taken as an input dataset and are listed in Table I.

TABLE I: INPUT DATASET

| Link ID | URL |
|---------|---|
| L1 | https://en.wikipedia.org/wiki/Computer |
| L2 | https://simple.wikipedia.org/wiki/Computer |
| L3 | https://www.webopedia.com/TERM/C/computer.html |
| L4 | https://www.gcflearnfree.org/computerbasics/what-is-a-computer/1/ |
| L5 | http://ecomputernotes.com/fundamental/introduction-to-computer/what-is-computer |
| L6 | https://www.computerhope.com/jargon/c/computer.htm |
| L7 | https://www.bhphotovideo.com/c/browse/Computers-Solutions/ci/9581/N/4294542559 |

Performance evaluation of the proposed system is done based on classification context scenario. Precision, Recall, Accuracy and F1 - Score plays a major role in evaluating the system.

Interpretation of Performance Measures [10]

True Positives (TP) - It means that the value of actual class is yes and the value of expected class is also yes.

True Negatives (TN) - It means that the value of actual class is no and value of expected class is also no.

False Positives (FP) - When actual class is no and expected class is yes.

False Negatives (FN) - When actual class is yes but expected class is no.

Accuracy - Accuracy is the most instinctive performance measure and it is simply a ratio of correctly predicted observation to the total observations.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$\text{Precision} = \frac{TP}{TP+FP}$$

Recall - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

$$\text{Recall} = \frac{TP}{TP+FN}$$

F1 score - F1 Score is the weighted average of Precision and Recall.

$$\text{F1 Score} = \frac{2 * (\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})}$$

From the performance measure, it is clear that the accuracy of the proposed system is 96% which is higher than existing approaches.

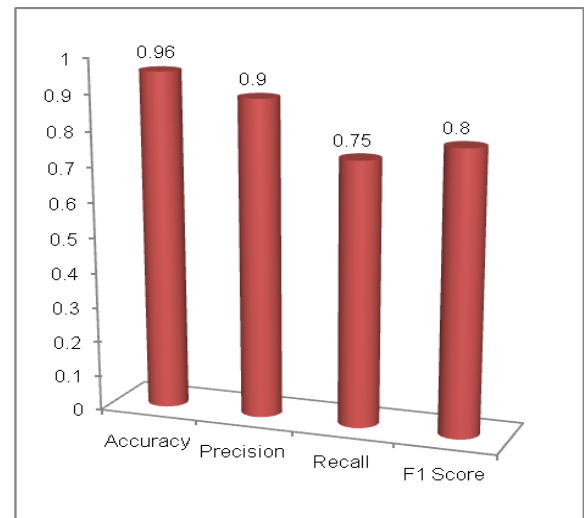


Fig 3: Performance of Proposed System

TABLE II. ACCURACY FOR DIFFERENT QUERIES

| Query | Existing System | Proposed System |
|-------------------|-----------------|-----------------|
| Computer | 96% | 97% |
| Grid Computing | 86% | 90% |
| Auto Mobile | 89% | 92% |
| Anti Virus | 91% | 94% |
| Algorithm | 90% | 96% |
| Research Paper | 92% | 92% |
| Projects for Kids | 90% | 91% |
| Page Rank | 89% | 95% |
| Data Mining | 94% | 95.5% |

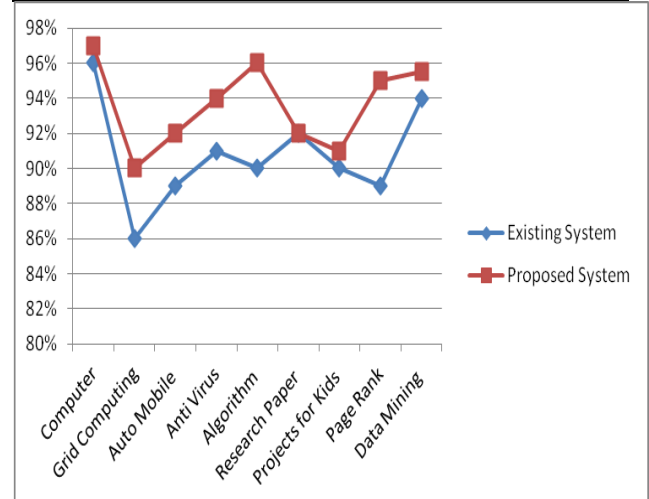
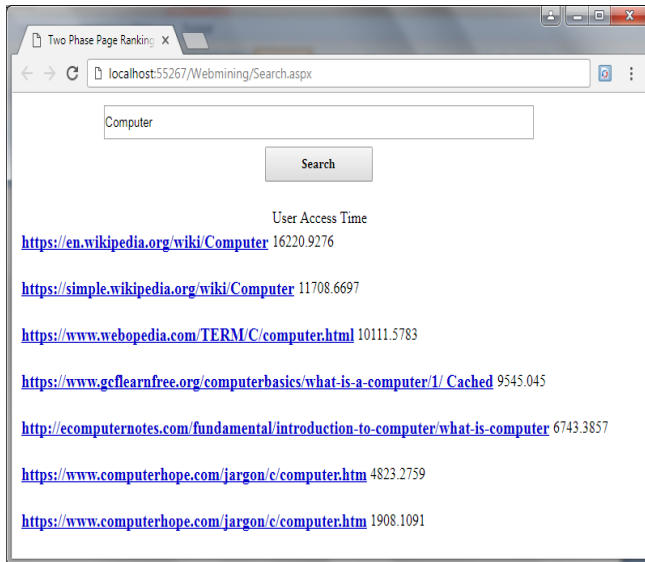


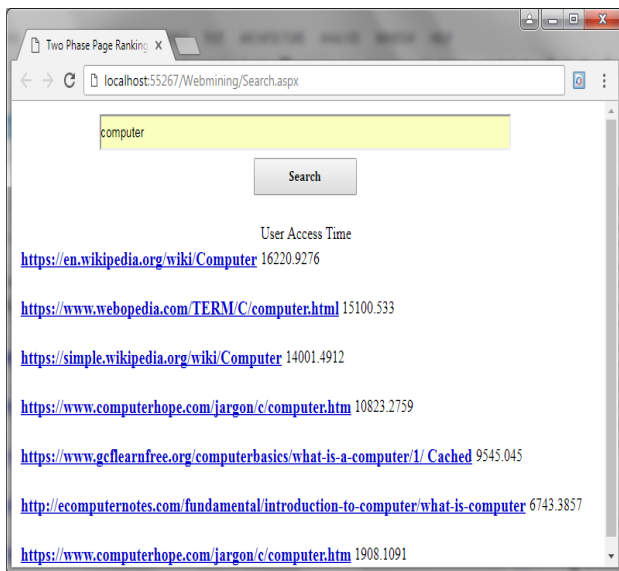
Fig 4: Comparison of Accuracy of Proposed System with Existing System

From Table II, it can be inferred that for first twelve links of rank list, the proposed algorithm gives more accurate results than existing content based algorithm.



| URL | User Access Time |
|---|------------------|
| https://en.wikipedia.org/wiki/Computer | 16220.9276 |
| https://simple.wikipedia.org/wiki/Computer | 11708.6697 |
| https://www.webopedia.com/TERM/C/computer.html | 10111.5783 |
| https://www.gcflernfree.org/computerbasics/what-is-a-computer/1/ Cached | 9545.045 |
| http://ecomputernotes.com/fundamental/introduction-to-computer/what-is-computer | 6743.3857 |
| https://www.computerhope.com/jargon/c/computer.htm | 4823.2759 |
| https://www.computerhope.com/jargon/c/computer.htm | 1908.1091 |

Fig 4: Result of TPPR for the Query “Computer” with user access time



| URL | User Access Time |
|---|------------------|
| https://en.wikipedia.org/wiki/Computer | 16220.9276 |
| https://www.webopedia.com/TERM/C/computer.html | 15100.533 |
| https://simple.wikipedia.org/wiki/Computer | 14001.4912 |
| https://www.computerhope.com/jargon/c/computer.htm | 10823.2759 |
| https://www.gcflernfree.org/computerbasics/what-is-a-computer/1/ Cached | 9545.045 |
| http://ecomputernotes.com/fundamental/introduction-to-computer/what-is-computer | 6743.3857 |
| https://www.computerhope.com/jargon/c/computer.htm | 1908.1091 |

Fig 5: Result of TPPR for the Query “Computer” after user access time of web pages increased

User Access time is a vital aspect in the TPPR algorithm. Using this, User can get the dynamic ranking of a particular web page for a query. Our search word is “Computer”. We have given first seven links with their access time for this keyword in Fig 4. From Fig 4, we can see that the link “[https:// www.webopedia.com/ TERM/C /c omputer.html](https://www.webopedia.com/TERM/C/c omputer.html)” has get 3rd rank in the list. However, it may get higher ranking position in the rank list if this page become accessed by the users. A web page becomes popular if the page gets many users to visit. Suppose, many users have visited the link “<https://www.webopedia.com/TERM/C/computer.html>” therefore the page rank of that page is increased. In Fig 5, we can see that the access time of the link “<https://www.webopedia.com/TERM/C/computer.html>” is increased and it gets the second position in the rank list.

TABLE III. COMPARISON OF EXISTING AND PROPOSED SYTEM ON THE BASIS OF VARIOUS FACOTRS

| Parameter / Technique | Existing System | Proposed System |
|--------------------------|--------------------|-----------------|
|--------------------------|--------------------|-----------------|

| Algorithm | Relevancy and Weight based approach | TPPR |
|-------------------------------------|---|------|
| Support of User Interest on Page | No | Yes |
| Time Efficiency | 87% | 97% |
| Accuracy | 82% | 96% |
| Sequence of Web Pages | 83% | 97% |

V. CONCLUSION

The Proposed approach produces extreme better results compared with search engine ranking. In this approach content mining and usage mining have been used. If the user visits a particular page frequently, and spend more time then the page will get higher rank. Moreover, this is a dynamic algorithm. If we add web structure mining with this algorithm, it may produce better result. Therefore, in future we will have the structure mining also as a feature in the approach. We conducted various experiments to evaluate the performance of TPPR and our findings are as follows:

- The proposed method works better than the existing algorithm. It provides most relevant in top of the result page. The order of the URLs is satisfactory.
- Title and Meta tags contribute a vital role in content extraction.
- User interest on the page acts as a major factor for computing web page ranking.

REFERENCES

1. P.Sudhakar, G.Poonkuzhali, R.Kishore Kumar. Content Based Ranking for Search Engines. Proceedings of the International MultiConference of Engineers and Computer Scientists 2012 Vol I. IMECS 2012, March 14-16,2012, Hong Kong.
2. M. Shamiul Amin, Shaily Kabir, Rasel Kabir. A Score based Web Page Ranking Algorithm. International Journal of Computer Applications (0975 – 8887) Volume 110 – No. 12, January 2015
3. Jayendra singh Chouhan, Anand Gadwal. Improving Web Search User Query Relevance using Content Based Page Rank. IEEE International Conference on Computer, Communication and Control (IC4-2015).
4. Ankita Kusmakar and Sadhna Mishra. Web usage Mining: A Survey on Pattern Extraction from Web Logs. International Journal of Advanced Research in Computer Science and Software Engineering. Volume 3, Issue 9, September 2013.
5. V.Lakshmi Praba and T. Vasantha.Evaluation of Web Searching Method Using a Novel WPRR Algorithm for Two Different Case Studies. ICTACT Journal on Soft Computing, April 2012, Volume: 02, Issue: 03
6. Madhurdeep Kaur and Chanranjit Singh. A Hybrid Page Rank Algorithm : An Efficient Approach. Internation Journal of Computer Applications Volume 100-No 16.August 2014.
7. PAN EI SAN. Main Content Extraction from Dynamic Web Pages. International Journal of Advances in Electronics and Computer Science, ISSN: 2393-2835 Volume-2, Issue-3, March-2015.
8. Najlah Gali and Pasi Fränti. Content-based Title Extraction from Web Page. In Proceedings of the 12th International Conference on Web Information Systems and Technologies (WEBIST 2016) - Volume 2, pages 204-210.
9. R.Gunasundari and Dr.S.Karthikeyan. A Study Of Content Extraction From Web Pages Based On Links. International Journal of Data Mining & Knowledge Management Process (IJDMP) Vol.2, No.3, May 2012.

10. <http://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures>
11. N. V. Pardakhe, Prof. R. R. Keole. Analysis of Various Web Page Ranking Algorithms in Web Structure Mining. International Journal of Advanced Research in Computer and Communication Engineering Vol.2, Issue 12, December 2013