IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

KNOWLEDGE BASED ADVISORY SYSTEM TO ANALYZE SEARCH THROUGH DIFFERENT PAGE RANKING IMPLEMENTATIONS

Manikandan & B V A N S S Prabhakar Rao
School of Computer Science and Engineering
Vellore Institute of Technology, Chennai.

ABSTRACT

The usage of gadgets and the web has seen an outburst growth and since internet accessibility has become widespread, the need for information retrieval from the web has become an integral area of focus. Presently search engines like Google have a drastic improvement in performance through the usage of Artificial Intelligence. The web is rich in information to the user and all websites communicate through sets of links and hyperlinks. Since searches provide us with exploration of information, it is necessary that the content of the web pages which users are getting in the search results is up to date, accurate and relevant to their needs. Focusing on the searches that provide information about a specific content, the sites should not provide misleading information. For dealing with the searches like disease symptoms, the users have to be given correct and up to date information. Since pages on the basis of a specific content are innumerable and finding the necessary content manually is impossible, the pages have to be ranked in terms of links directing to the webpage and of those moving out of it. For implementing this we have the PageRank algorithm which is primarily applied by search engines like Google, for providing the users with necessary, accurate and updated content based on their searches provided. Ranking of web pages is done on the basis of in-links and out-links of it. Ranking is done to identify the most relevant resources with the highest quality among all the relevant resources on the web and other information retrieval systems. The main motive behind PageRank is to provide the relative importance of a page. The PageRank algorithm provides the output as a probability distribution that is used to represent the possibility that a user will arrive at the page through directing and redirecting links. There are different versions of page ranking algorithms and each implementation aims to optimize the idea of the page ranking algorithm. In this paper we focus on the analysis of different page ranking algorithms to provide best optimization for searches so that the users get accurate information regarding search results.

Keywords – Artificial Intelligence, Google Search, Links, Hyperlinks, Webpage Content, PageRank, Search Optimizations

INTRODUCTION

Searching the web has seen an utmost popularity in the current decade and for the current generation that is dependent on computers and gadgets, it is important for search engines to have maximum possible accuracy and relevancy in their content [22]. The search engine has to provide accurate, up-to-date and relevant content of the input keyword provided by the user. Focusing on the content searches, the pages that are provided by the search engine as a response to the searched keyword have to be accurate [4], and the search engine should ensure that

pages with incorrect and outdated content, although relevant to the keywords, aren't displayed in the search results page [6]. The PageRank algorithm provides a probability distribution as the output that provides the chances that a user arrives at that particular page through in-links and out-links of the page [1]. Following explains the architecture and working of the search engine [9].

Architecture and Working of the Search Engine:

1. Crawler - This is a module that is used to collect the web pages. The pages that are collected by the crawler are known as crawled web pages.
2. Indexer - A module that creates inverted files based on the crawled web pages [14].
3. Based on a generated keyword web map is generated
   -- User enters a query in a search box.
   -- Query servers generate Keywords from the text entered by the user [16].
4. Ranked engine generates rank for each webpage.
5. Generate web page result sets.

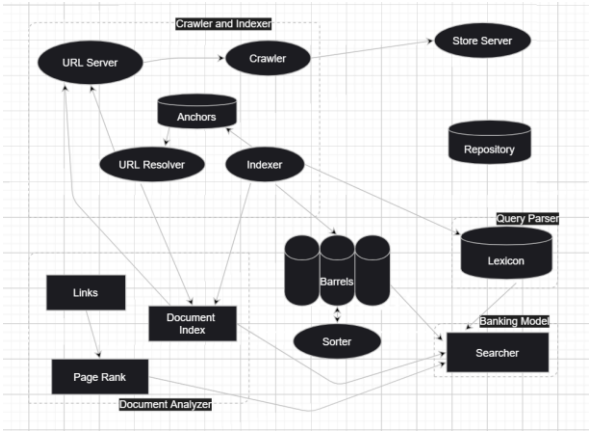
Crawling, Ranking and Indexing are the essential parts of the search engine [25].
The working of a Search engine is as follows:

- ❖ Firstly, the search engine performs the operation of collecting the pages from the web and storing them in a repository, known as crawling.
- ❖ Secondly, it analyzes the pages in the repository and extracts the title and link (URL). From the major parts of the pages, keywords are gathered, which are known as search terms.
- ❖ Then Ranking is performed which involves the method of calculation of the rank of the pages which portrays the importance of the crawled page.

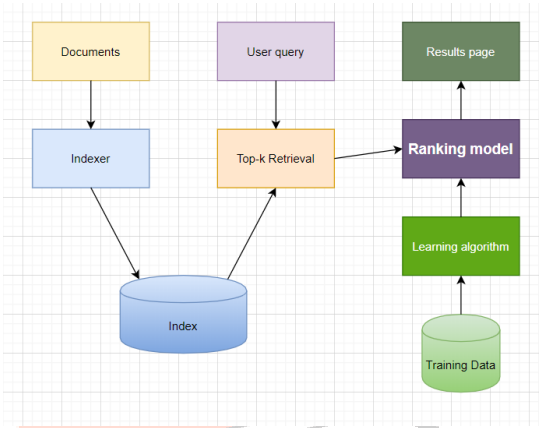
Ranking means identifying the most relevant resources with the highest quality among all the relevant resources on the web and other information retrieval systems [7].

PageRank is an evolutionary Ranking algorithm and since 1998, it is implemented by Google web search engine.

Google Architecture



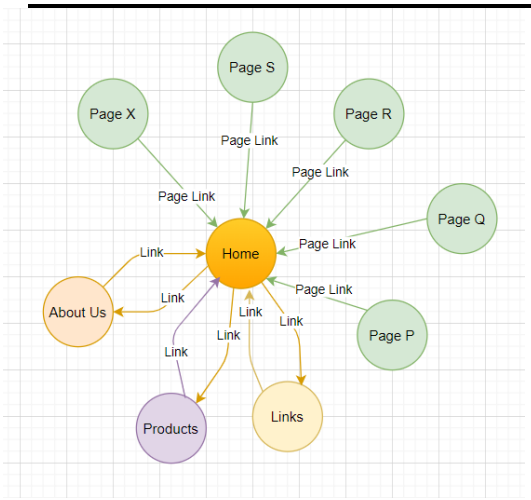
Architecture Diagram of a Ranking system:

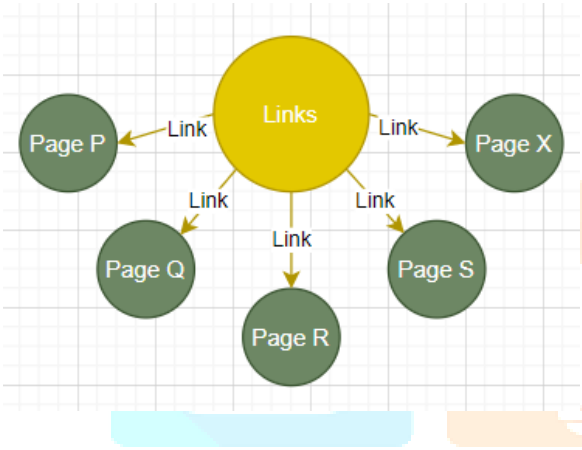


BACKGROUND STUDY AND RELATED ALGORITHMS

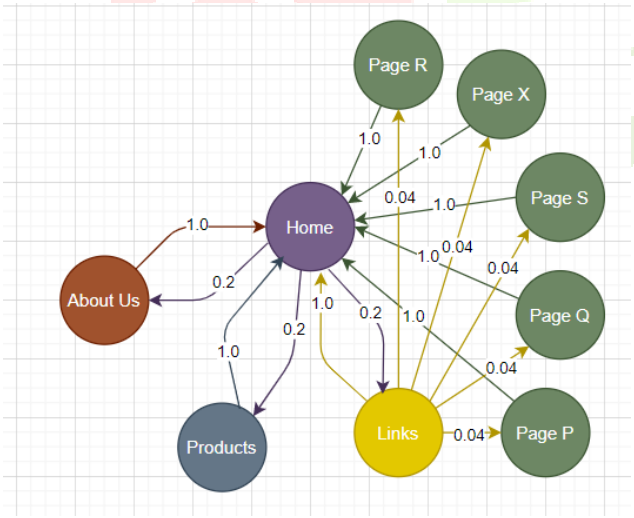
Algorithm #1 - PageRank Algorithm:

PageRank is a link analysis algorithm which assigns decimal weight values to each element of a hyperlinked set of documents, such as the WWW, with the purpose of analyzing its relative importance within the set. The algorithm's application tends to any collection of entities with reciprocal quotations and references. The numerical weight that is assigned to a given element E is termed as the PageRank of E and is denoted by PR(E).
A PageRank is resulted from the algorithm based on the web-graph, created by all WWW pages as nodes and hyperlinks as edges.
Below graph shows a typical example of a PageRank Algorithm.





Here's the example graph of PageRank but with weighted values.



Algorithm:

```
procedure PageRank(G, iterations):
    dampingFactor ← 0.85
    outHash ← G
    inHash ← G
    NumberofPages ← G
    for all page in the graph do
            oldPageRank[p] ← (1/NumberofPages)
    end for
    while iterations > 0 do
            pageDampingFactor ← 0
            for all page that has no out-links do
                    pageDampingFactor ←
pageDampingFactor + dampingFactor *
(oldPageRank[page] / NumberofPages)
            end for
            for all page in the graph do
                    newPageRank[page] ←
pageDampingFactor + ((1-dampingFactor)
/NumberofPages)
                    for all inPage in inHash[p] do
                            newPageRank[page] ←
newPageRank[page] + (dampingFactor *
oldPageRank[inPage] / outHash[inPage])
                    end for
            end for
            oldPageRank ← newPageRank
            iterations ← iterations - 1
    end while
end procedure
```

Algorithm #2 - Hyperlink-Induced Topic Search (HITS) algorithm:

Firstly, the algorithm involves selecting and getting the pages that are most relevant, and send to the search query. The primary set is the root set, which can be gathered by selecting the top-rated pages that is returned through a text-based search algorithm. By exacerbating the root set with all the pages linked from it (out-links), and link to it (in-links), the base set can be obtained. The pages present in the base set and all links constitute the focused subgraph, where the computation of the algorithm is implemented. It consists of a series of iterations each having these important increment operations:

- Increment each node's authority score as the sum of each hub scores pointing to it, that is, a node is given a high authority score by linking from pages that are considered as information Hubs.
- Increment each node's hub score as the sum of the hub scores of each node pointed to, that is, a node is provided with a high hub score by linking to nodes of subject authorities.

The hub's and authority's score is assigned by using the following algorithmic steps:

1. Start with hub and authority score of 1 for each node.
2. Execute the respective hub and authority update steps.
3. By finding the mean square values of the hub and authority scores, normalize the values.

4. Repeat second step when necessary.

Difference from PageRank:

- It is dependent on query, i.e., the link analysis scores are influenced on the search items.
- It is executed at the time of query, with the associated performance hit that accompanies processing time of queries.
- It is not commonly used by search engines.
- It computes two scores per page.
- It is processed on a small subset of relevant pages, not all pages as the case with PageRank.

Pseudocode:

```
Let G be the set of pages.
for each page Page1 in G do
   Page1.authority = 1;
   Page1.hub = 1;
   for step from 1 to n do
            normval = 0;
            for each page Page1 in G do
            p.authority = 0;
            for each page Page2 in
Page1.incomingNeighbours do
                     Page1.authority += Page2.hub;
            normval += square(Page1.authority);
   normval = squareroot(normval);
   for each page p in G do
   Page1.auth = 0;
   for each page Page2 in Page1.incomingNeighbours do
            Page1.authority += Page2.hub;
   normval += square(Page1.authority);
   normval = squareroot(normval);
   for each page Page1 in G do
            Page1.hub= (Page1.hub)/normval
```

Algorithm #3 – Improved HITS (I-HITS) algorithm:

HITS algorithm is symmetric, in the sense that both hub and authority weights are defined in the same way. The algorithm is also egalitarian, i.e., when computing the authority weight of some page p, the hub weights of the pages that point to page p are all treated equally. However, these properties of the algorithm may lead to non-intuitive results.

If a page, say page X points to another page, page Y, then X is known as the source page and Y is known as the target page. The more popular a page is, the more other pages tend to point to or it will be linked to by other pages. The experiment uses more specific and detailed ternary evaluation and classifies a document as:

1 – Highly Relevant (HR): This evaluation contains much essential and authoritative information about the given query.

2 – Relevant (R): Has relevant but not necessary information about the given query.

3 – Non-Relevant (NR): Includes neither the keywords of the given nor relevant information about it.

For each page, the count of each category is compared and the category with the largest count is selected.

The more popular a page is, the more the other pages tend to point to it or it will be linked to other pages. The proposed version allocates bigger ranks to more essential pages instead of partitioning the rank of a page evenly among its out-link pages.

From the results, it is concluded that the I-HITS algorithm that is proposed as an improvement over the HITS algorithm, has increased the ability to distinguish the link importance of the page, and avoids top drift. Comparatively, the improvement is better when searching related pages with an increment in query quality.

Algorithm Analysis Table:

Algorithm	Functions	Features
PageRank	Link Analysis algorithm Assigns decimal weight values to hyperlinked pages Analyzes relative importance within the set	Finds the relevancy in pages through ranking Search optimization is done through ranking of pages
HITS	Consists of authority and hub scores for each page. Performing respective hub and authority score updates and normalize the values through mean square values.	Query dependent Execution at the time of query. Computes two scores per page. Processes on a small subset of relevant pages
I-HITS	Improvement over HITS; Detailed ternary evaluation Assigns three classifications, Highly Relevant, Relevant and Non-Relevant. Count of each category is compared. Allocates bigger ranks to more essential pages.	Increased ability to distinguish link importance of a page. Avoids top drift. Better improvement when searching related pages with increment in query quality.

Comparison of PageRank and Weighted PageRank:

Analyzing Factor	PageRank	Weighted PageRank
Definition	Link-analysis algorithm considering only backlinks.	Link-analysis algorithm focusing on weighted values of in links and outlinks.
Concept of Web Mining Applied	Web Structure Mining	Web Structure Mining
Time Complexity of Algorithm	O(log(n))	Less than O(log(n))
Query Dependency	Independent of query	Independent of query
Explanation	Calculates page ranks at indexing time by evenly distributing rank values among outlinked pages.	Calculates page rank weighted values at indexing time by unevenly distributing rank values among outlinked pages.
Quality of Outcome	Medium Quality	Medium Quality but higher than traditional PageRank
Relevancy	Less relevant since the algorithm works at indexing time.	Less relevant since the algorithm calculates weights at indexing time.
Merit	Rank calculation is done based on page importance.	Assigns larger weights to more important pages.
Demerit	Favours older and outdated pages since rank is calculated based on page links.	Works only based on web page popularity.

Comparison of HITS and PageRank:

Analyzing Factor	PageRank algorithm	HITS algorithm
Definition	Link analysis algorithm based on random surfer model	Link analysis algorithm
Web Mining technique	Web Structure Mining	Web Content Mining and Web Structure Mining
Functionality	Computes rank at crawling time. A combined rank with information retrieval score is analyzed. More efficient.	Invokes traditional page sets of SEs and finds it's hub and authorities. Computed at query time. Not feasible for SEs.
Mutual Reinforcement	Doesn't distinguish between the hub and the authority of a page. Just calculates page rank based on authority of the page.	Emphasizes between authority and hubs.
Query Dependency	Query Independent	Query Dependent
Algorithm Stability	Unstable. Values may vary drastically if links are modified.	Unstable. Scores change based on link modification.
Input Parameters	Backlinks	Content, Frontlinks and Backlinks
Neighbourhood Applications	Applied to the entire web	Applied to the local neighbourhood of pages based on query results.

METHODOLOGY

User Side:

As soon as the users install the extension, they can start highlighting any content in the webpage. When they highlight a content, a tooltip appears with two buttons, Useful, shown in green, and Inaccurate, shown in red. If the user clicks either button, the content selected is highlighted and is automatically sent to the database. The particular URL is also saved for score analysis. A positive score is sent if the user has clicked Useful and a negative score is sent if the user has clicked Inaccurate.

'overallScore' variable is calculated based on the sum of scores of the URL, and if the page has received positive response more, then the link is highlighted green on the

Google Search Page. If the page has received more negative responses, then the link is highlighted red on the search page. In case both positive and negative responses are equal, then the link is highlighted gray on the search page.

Developer Side:

The first step is recording all highlights made by the users and retaining the highlights on the page. This is done by storing all recorded highlighted contents on the database. The respective URL also is saved when the user provides a response. The backend takes care of the tag results provided by the user and respectively records it in the database. After saving the URL, the backend evaluates the overall score of the URL from the saved scores on the content in the URL.

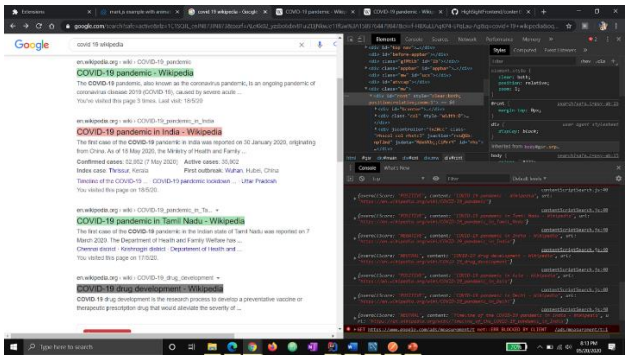
The frontend receives the overall score from the backend and respectively highlights the URL based on the nature of the overall scores received. For overall positive response, the link is highlighted Green, for negative response the link is highlighted Red, and Gray highlight is done for neutral response.

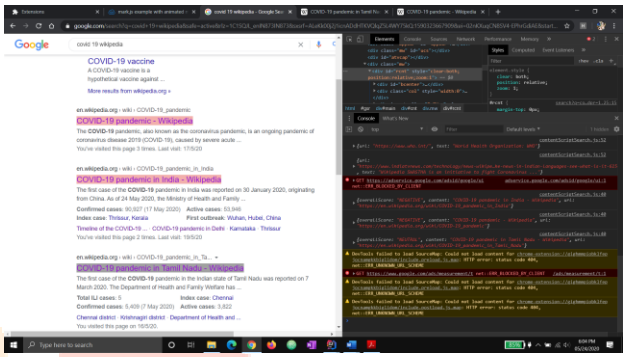
The frontend of the project is done in JavaScript. Google Chrome extensions are made using three main files, manifest.json, background.js and contentScript.js. The manifest file initiates all the files and the processes of the chrome extension. The background JavaScript file takes care of initiating all event listeners sent by the contentScript file. The contentScript file is where the desired code is written.

The backend and server are implemented in Java. I have partitioned the backend into three categories, controller, repo and service. Since highlights and the page have to be separately evaluated by the backend, each of both the Highlights and Pages classes have the three categories of files:

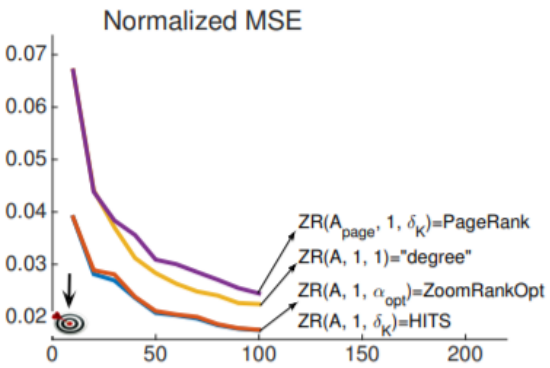
- The files in the controller partition are responsible for receiving the highlight request, assigning tags and generating respective score.
- The repo partition is responsible for assigning the respective score key for the highlight content and interactions with the database.
- The service is responsible for calculating the overall score and interactions with the frontend after evaluation of overall score.

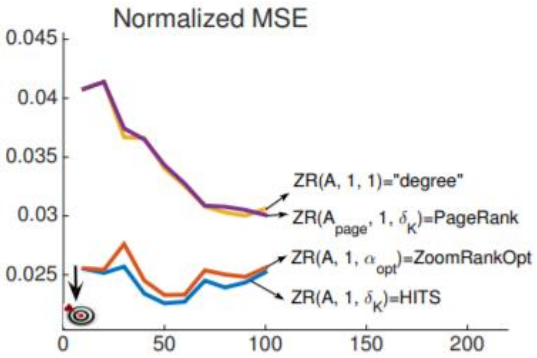
IMPLEMENTATION & DISCUSSION





Comparison Graphs of PageRank and HITS:





The main aim of this implementation is to show the manual ranking of the page links by accepting user highlights. Finding Promoted Content is the main motive of the Search Engine Optimization. Search Engines always aim for increasing the rank of the promoted content.

CONCLUSION

Hence each version of page ranking algorithm's functionalities, improvements and working is analyzed theoretically. The difference between PageRank and HITS algorithm is studied and the other two improvement algorithms have been analyzed and the area of improvement is found through the study of the algorithm. Each of the algorithm has its own advantages and disadvantages, hence the search engine has to analyze the merits and demerits of the algorithms to make optimized searches. As discussed before, since the popularity of searches is high, optimization is a key factor that has to be focused.

REFERENCES

[1]: T. Sen, D. K. Chaudhary and T. Choudhury, "Modified Page Rank Algorithm: Efficient Version of Simple Page Rank with Time, Navigation and Synonym Factor," *2017 3rd International Conference on Computational Intelligence and Networks (CINE)*, Odisha, 2017, pp. 27-32.
doi: 10.1109/CINE.2017.

[2]: S. Yerma and A. K. Majhvar, "Updated page rank of dynamically generated research authors' pages: A new idea," *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, 2016, pp. 879-882.
doi: 10.1109/RTEICT.2016.7807954

[3]: M. Usha and N. Nagadeepa, "Combined two phase page ranking algorithm for sequencing the web pages," *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, Coimbatore, 2018, pp. 876-880.
doi: 10.1109/ICISC.2018.8398925

[4]: Divjot and J. Singh, "Effective Model and Implementation of Dynamic Ranking in Web Pages," *2015 Fifth International Conference on Communication Systems and Network Technologies*, Gwalior, 2015, pp. 1010-1014.
doi: 10.1109/CSNT.2015.261

[5]: L. Smitha and S. S. Fatima, "Topical and Trust Based Page Ranking Using Automatic Seed Selection," *2017 IEEE 7th International Advance Computing Conference (IACC)*, Hyderabad, 2017, pp. 803-806.
doi: 10.1109/IACC.2017.0165

[6]: M. P. Selvan, A. C. Shekar, D. R. Babu and A. K. Teja, "Efficient ranking based on web page importance and personalized search," *2015 International Conference on Communications and Signal Processing (ICCSP)*, Melmaruvathur, 2015, pp. 1093-1097.
doi: 10.1109/ICCSP.2015.7322671

[7]: Nagappan V.K and P. Elango, "Agent based weighted page ranking algorithm for Web content information retrieval," *2015 International Conference on Computing and Communications Technologies (ICCCT)*, Chennai, 2015, pp. 31-36.
doi: 10.1109/ICCCT2.2015.7292715

[8]: J. Xia, Y. Hou, Y. V. Chen, Z. C. Qian, D. S. Ebert and W. Chen, "Visualizing Rank Time Series of Wikipedia Top-Viewed Pages," in *IEEE Computer Graphics and Applications*, vol. 37, no. 2, pp. 42-53, Mar.-Apr. 2017.
doi: 10.1109/MCG.2017.21

[9]: R. Singhal and S. R. Srivastava, "Enhancing the page ranking for search engine optimization based on weightage of in-linked web pages," *2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, Jaipur, 2016, pp. 1-5.
doi: 10.1109/ICRAIE.2016.7939544

[10]: V. V. Mahale, M. T. Dhande and A. V. Pandit, "Advanced Web Crawler For Deep Web Interface Using Binary Vector & Page Rank," *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on*, Palladam, India, 2018, pp. 500-503.

[11]: L. Rodrigues and S. Jaswal, "Hybrid model for improvised page ranking algorithm," *2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, Kumaracoil, 2015, pp. 466-469.
doi: 10.1109/ICCICCT.2015.7475324

[12]: S. K. Guha, A. Kundu and R. Dattagupta, "Web page ranking using domain-based knowledge," *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Kochi, 2015, pp. 1291-1297.
doi: 10.1109/ICACCI.2015.7275791

[13]: N. Jain and U. Dwivedi, "Ranking web pages based on user interaction time," *2015 International Conference on Advances in Computer Engineering and Applications*, Ghaziabad, 2015, pp. 35-41.
doi: 10.1109/ICACEA.2015.7164709

[14]: A. Gupta, A. Dixit and P. Devi, "A novel user preference and feedback based Page Ranking technique," *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, 2015, pp. 1335-1340.

[15]: Anjali, A. Sadhwani and N. Saxena, "A new approach to ranking algorithm - Custom Personalized Searching," *2015 2nd International Conference on Computing for*

Sustainable Global Development (INDIACom), New Delhi, 2015, pp. 130-133.

[16]: J. Singh Chouhan and A. Gadwal, "Improving web search user query relevance using content-based page rank," *2015 International Conference on Computer, Communication and Control (IC4)*, Indore, 2015, pp. 1-5. doi: 10.1109/IC4.2015.7375680

[17]: F. Zhan *et al.*, "An efficient alternative to personalized page rank for friend recommendations," *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, 2018, pp. 1-2. doi: 10.1109/CCNC.2018.8319307

[18]: H. Chu, C. Yan, Z. Luo and X. Huang, "The Improvement of Web Page Ranking on SERPs," *2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, Taichung, 2018, pp. 1-2. doi: 10.1109/ICCE-China.2018.8448460

[19]: S. Krrabaj, F. Baxhaku and D. Sadrijaj, "Investigating search engine optimization techniques for effective ranking: A case study of an educational site," *2017 6th Mediterranean Conference on Embedded Computing (MECO)*, Bar, 2017, pp. 1-4. doi: 10.1109/MECO.2017.7977137

[20]: M. K. Mittal, N. Kirar and J. Meena, "Implementation of Search Engine Optimization : Through White Hat Techniques," *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, Greater Noida (UP), India, 2018, pp. 674-678, doi: 10.1109/ICACCCN.2018.8748337.

[21]: K. Nepomuceno, T. Nepomuceno and D. Sadok, "Measuring the Internet Technical Efficiency: A Ranking for the World Wide Web Pages," in IEEE Latin America Transactions, vol. 18, no. 06, pp. 1119-1125, Jun 2020, doi: 10.1109/TLA.2020.9099750.

[22]: M. Butkiewicz, H. V. Madhyastha and V. Sekar, "Characterizing Web Page Complexity and Its Impact," in IEEE/ACM Transactions on Networking, vol. 22, no. 3, pp. 943-956, June 2014, doi: 10.1109/TNET.2013.2269999.

[23]: S. Rodriguez-Vaamonde, L. Torresani and A. W. Fitzgibbon, "What Can Pictures Tell Us About Web Pages? Improving Document Search Using Images," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 6, pp. 1274-1285, 1 June 2015, doi: 10.1109/TPAMI.2014.2366761.

[24]: J. Kim, "A Document Ranking Method With Query-Related Web Context," in IEEE Access, vol. 7, pp. 150168-150174, 2019, doi: 10.1109/ACCESS.2019.2947166.

[25]: V. N. Gudivada, D. Rao and J. Paris, "Understanding Search-Engine Optimization," in Computer, vol. 48, no. 10, pp. 43-52, Oct. 2015, doi: 10.1109/MC.2015.297.