Advances in Mathematical & Computational Sciences Journal,
Vol. 7 No. 1, March 2019

Article Citation Format
Naibi, R. S., Atiku, A.U. & Sajoh, D.I. (2019): On Search Engines and Web Page Ranking Algorithms.
Journal of Advances in Mathematical & Computational Sciences
Vol. 7, No. 1. Pp 41-50

Article Progress Time Stamps

Article Type: Research Article
Manuscript Received 13th Jan. 2019
Review Type: Blind
Final Acceptance: 17th March, 2019
Article DOI: dx.doi.org/10.22624/AIMS/MATHS/V7N1P6

On Search Engines and Web Page Ranking Algorithms

[1]Naibi, R. S., [2]Atiku, A.U. & [3]Sajoh, D.I.
Department of Computer Science
Modibbo Adama University of Technology
Yola, Adamawa State, Nigeria
E-mails: [1]ridwan.naibi@mautech.edu.ng, [2]ahmed.atiku@mautech.edu.ng, [3]disajoh@mautech.edu.ng

ABSTRACT

This paper presents a review of search engines, and the current State of the Art technologies in use by popularly known search engines. This paper does not attempt to report in all the technical details of search engines as some details are proprietary technology and business secrets. We avoided including search engines that are not available for use around the globe or those that were specialised for particular languages or dialects in our study.

Keywords: Search Engines, Web page, Ranking, Algorithms, Optimization and technologies.

1. INTRODUCTION

Search engines are Information Retrieval Systems used to extract the required information by a user usually through a specified query string (Kleinberg, 1999). Search engines are very important Internet tools as they make information retrieval very tractable as the web has grown at an astronomical rate since inception in 1990 (Seymour et. al., 2011). In fact, at this present time, without search engines, information retrieval or search on a certain topic is impossible except if one knows exactly the site to follow to get the information that one requires. Practically speaking, because of the size of the web and the nature of information search, a list of sites to be accessed for information cant be kept because in most times where the required information one is searching for is not known, and even if a list is kept, the list becomes intractable because it grows very fast even for the average internet user.

1.1 Review Method

The review method below has been adapted from the evidence-based software engineering Systematic Literature Review Template (Kitchenham et. at., 2015). This Systematic Literature Review aims to extract information that will guide in building a search engine with three different link analysis ranking algorithms

(PageRank, HITS, and SALSA). The topics to be scoped are search engines, and the link analysis ranking algorithms.

2.2 Search Strategy

The search strategy was based on the search of electronic databases and a scholarly search engine namely; Web of Science (also known as Web of Knowledge), IEEEXplore, Science Direct, Google Scholar search engine, and Durham University library catalogue. The following keywords were constructed and used for searching the aforementioned databases and search engines in this review:

1. Search engines
2. PageRank
3. Hyperlink induced topic search algorithm
4. Hubs and Authorities
5. Stochastic approaches to link-structure analysis

Selection Criteria: The inclusion criteria were: All scholarly or scientific sources relating to search engines will be included, and source must not be a duplicate. The exclusion criteria were: All the sources in form of presentations (PowerPoint etc.), informal discussions, those not found through scholarly mediums will be excluded.

Data Extraction: A list of sources that may be considered is generated. After reading the title, keywords, and abstracts of the generated list of papers and comparing them to the scope of the study, relevance to the research topic and inclusion and exclusion criteria, a source is either included or excluded, reasons for inclusion is stated, then paper is categorized. A no duplicity check is enforced on the paper and comments sections of the classification table using excel to make sure no paper is entered twice.

Synthesis: The papers generally addressed search engines, and the different link structure analysis algorithms, which are HITS, PageRank, and SALSA

Threats to validity: There are no visible threats to validity as sources were selected based on relevance to the research question and how sufficiently they can be used to address the research question, which is basically to guide the implementation and testing of deliverables that would help answer the research questions

Limitation: The papers were not tested for quality of information they present but for their effectiveness in helping address the research question.

2. REVIEW

2.1 History of Search Engines

Search engines are basically machines used for looking up information on the web, and also referred to as information retrieval systems (Lempel and Moran, 2000). Search engines usually look through their databases sometimes referred to as an indexer to quickly determine what information the searcher is requesting (Seymour et. al., 2011). Search engines normally employ a two-step process, which is looking up documents through word processing and also by scouting for documents that have to do with the semantic meaning of the supplied search string (Langville and Meyer, 2004, pp. 335–380).

According to Seymour et. al (2011) during the early development of the web, there was a list of web servers edited by Tim Berners-Lee and hosted on the CERN web server. As more web servers went online the central list could not keep up. On the NCSA site new servers were announced under the title "What's New!" but could not keep up with the astronomical birth of new web servers. This in turn necessitated the need for search engines. Although, there have been a lot of developments on search engines, a synopsis of a few remarkable ones from Seymour et. al (2011) is provided below:

- **Archie Search Engine:** "Archie" was the first search engine created in 1990 by Alan Emtage, Bill Heelan and J. Peter Deutsch who were at the time computer science students at McGill University in Montreal. "Archie" keeps a list of all FTP (File Transfer Protocol) sites by creating a manually searchable database. "Archie" is operated by UNIX commands
- **AltaVista - (1995):** AltaVista was the most popular search engine with a very powerful server which could survive millions of information retrievals per day without crashing right before the advent of Google, it was built by Louis Monier(built the web crawler), and Michael Burrow(built the indexer).
- **WebCrawler - (1994):** Brian Pinkerton, a computer science student at the University of Washington, built the WebCrawler. It was the pioneer of full text search and went live with over 4000 distinct Web sites on April 20, 1994. It was the first search engine that had an indexer for all words of a web page while others only indexed titles, URLs and a limited number of words
- **MSN Search- (2005):** MSN Search (now known as Bing) a search engine proprietary to Microsoft, which has an indexer, and web crawler. The interesting thing with MSN Search is, it provided image searching from a third-party search engine known as Picsearch.
- **MetaCrawler - (1995):** MetaCrawler is an offset of Search Savvy (which allows searching of up to 20 search engines through a single interface using one or more directories) created by Daniel Dreilinger while at Colorado State University. MetaCrawler is more efficient than Search Savvy because it has its own set of custom search syntax, which transforms the search queries to match the respective search engines to be queried for results.
- **Google (1998):** Google successfully reengineered the way search engines work through its proprietary PageRank algorithm, which is a ranking algorithm for ordering the search results based on the importance of each retrieved page. This new approach to ranking search result was designed by exploiting the inherent nature of the web which can be modeled as directed graphs, and then link analysis can be carried on the directed graph. PageRank which ranks pages based on the links a certain page has to itself from other pages with the notion that if important pages point to certain page then the pointed page itself is very likely to be an important page. PageRank will be discussed in detail later. It is widely believed that Google has added numerous other indicators to computing PageRank of pages, which are trade secrets of Google. Google's indexing is very interesting as it indexes all sorts of documents, including images, PDF files, and Word Document, Excel spreadsheets, Flash SWF and a host of others

2.1 Anatomy of Search Engines (Brin and Page, 1998)

- **Web Search Engines**: The size and the structure of the web presents us with a wealth of information and a big challenge. A very efficient quick crawling technology is required to store the web pages and make sure they are current. Efficient storages places are required to store the indices and most often the documents themselves. Therefore, indexing systems must have the capacity to process vast gigabytes of data in a very efficient manner i.e at order of thousands per seconds. The size of the web makes a difficult and challenging task but at the same time the link structure provide good information for probabilistic analysis. A two-pronged approach is therefore what an efficient search engine should use, taking advantage of the link structure and finding a workaround solution to the enormity of the available data on the web. Google was designed to scale the web efficiently by the use of indexes, reducing disk seeking (consequently reducing required time to run the seeking process), and storing compressed web data. Furthermore, the link structure in the light of a certain user's experience was exploited to measure what relevance and importance mean to a certain user.

- Systematic Features: Google uses two important features to improve search results i.e the link structure of the web to calculate the relevance and importance for each Web Page called PageRank, and links to improve Web results. We are not going to delve into the intricacies of the PageRank algorithm here, as we will revisit these in great detail later. PageRank represents the information discerning skills of the average user based on the notion of quality, which is read from users consecutive visiting of pages through link or a " jump" i.e directly going to random page by typing in the web address not clicking a link. This in turn guarantees that an average probabilistic model for a user can be learned, which in turn allows for personalization, which means the search engine can't be mislead through dubious activities like link farms – replicating links on random connected sites to improve the importance (PageRank) of a certain Web Page. Most importantly, some pages like Yahoo or CNN are considered cardinal because they are highly likely to only point to pages of high and qualitative information as they themselves are where the truest form of information may be found. Moreover, pointers may give us more information on the name and even when it holds a document that need not be indexed.

- System Components: We are going to avoid the "messy" details of underline technology here because of space constraints. There exists several web crawlers in order to achieve the inherent advantage of faster running time that is inherent with distributed systems. These crawlers pull-up web pages that are compressed, stored and efficiently tracked by the indexer. The indexer performs some computational "magic" to determine structure, capitalization, and occurrence (hits on words) in a document by creating a barrel, which is a form of storage for such computations, and consequently creating a considerably sorted index. The index also extracts information from anchor, which are links that give information to pages or contents where text analysis can't be used through building an anchors file. All links from the anchors files are then analyzed, stored and tacked just as other links and documents. From the storage of all documents, links, and anchor links, the PageRank is then computed. As mentioned before, several crawlers are used which raise challenges associated with interacting with other web pages, which may result into bugs that can only be fixed if web master's or the site where the bug is originating from contacts the operator of the crawler. The different parameters tracked by Google makes it impossible for any parameter to have too much influence in the computation of PageRank, which helps a great deal in forestalling tricking of the Google search engine.

- System Performance: Google has been able to prove its efficiency through the quality of search results it presents. Google can find pages that do not contain textual description or name of itself through its ingenious anchor link analysis. Most importantly. Emails, which are not crawlable, can also be searched because of anchor text and proximity tracking that Google does. Due to the many parameters used by Google to generate and rank search results, it doesn't return broken links. Especially for pages with a high PageRank. Google has been able to include economies of scale both in hardware cost and computational resource allocation by using distributional computing and minimizing resource allocation by only redoing indexing and crawling only sections of the Web that require such. Succinctly put, Google is very scalable searches engine whose primary design goals is to provide high quality search results for the astronomically growing World Wide Web. Google in its own right is architecture for crawling the Web, indexing web pages and running search queries to determine results. Other important features of Google is its ability to be personalized to a certain user in the light of the user's surfing activities, and usage as a research tool into the behaviors of surfers in order to improve search efficiency and human notion and judgment on returned results quality

3. RANKING ALGORITHMS

1*) PageRank*

In 1995, Larry Page and Sergey Brin doctoral students at Stanford University developed the idea of PageRank which led to their taking a leave of absence from doctoral study to startup a full-fledged search engine company named Google in 1998. As pointed out earlier PageRank exploits the hyperlink nature of the web, which is a collection of highly connected web pages through the form of link analysis. This forms link analysis took into account the in-links (links pointing to the page in question) and out-links (links the page in question is pointing towards). Interestingly, PageRank is query independent, and it was virtually protected to the earlier forms of spamming (Farahat et al., 2006).

The PageRank algorithm recursively computes the importance of each page based on the number of important links pointing to that page which to large extent correlates with human perception (Langville and Meyer, 2004). PageRank uses the hyperlinked structure of the web, by transforming into a directed graph and then representing it as a Markov Chain with a primitive transition probability Matrix. The probability matrix then forced to become irreducible which implies that a vector exists for the said matrix (Ding et al., 2002). From the hyperlink nature of the web, a Marko chain with a primitive transition probability matrix **P** is built. Consequently, when the chain is made irreducible it grantees that in the long run a stationery vector π^T exists, which is referred to as the PageRank Vector. If the power method is applied on the PageRank Vector, which is a primitive matrix, it will converge to a stationary vector (Langville and Meyer, 2004). The conversion rate of the power method is determined by the subdominant eigenvalue of the transition rate matrix (Langville and Meyer, 2004; Stewart 1994).

To show how the transition probability matrix built from the hyperlink structure of the web to become a stochastic and primitive one, we implore the example used by Langville and Meyer (2004) and some explanations offered by Farahat et al. (2006) and some by Ding et al. (2002). Supposing, the hyperlink web structure of the web is represented as a directed graph G with six nodes, the nodes of the graph represent the web pages while the directed arcs, which represents the hyperlinks, represents the probability of traversing from position *i* (page *i)* to position *j* (page *j)* in a single time step.

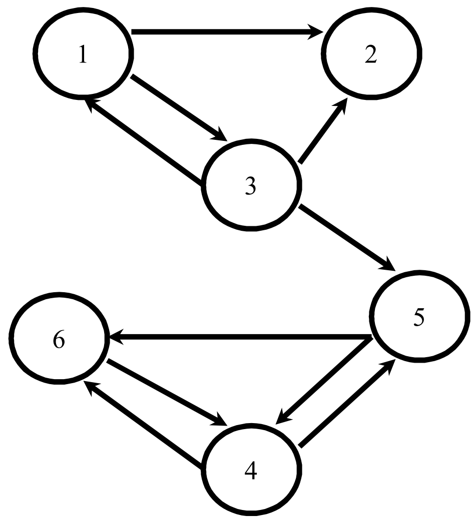

Figure 1: Directed graph modeling web page of 6 pages

From above the graph **G** as mentioned above we build a square matrix **P** that represents the Markov model whose element $\mathbf{P}_{ij}$. It is important to note that any suitable probability distribution can be used across all rows in the matrix with the assumption that is possible to start our transversal from any node and finish at any node.

$$\mathbf{P} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}.$$

The second row in **P** presents us with a new problem as a row consisting of all zeros tell us that **P** is not stochastic; which represents nodes that have no out-links referred to as dangling nodes, which many exits on the web. To make **P** stochastic we replace all 0^T with $1/n(e^T)$, where e^T represents the row vector consisting of only ones, and n is the order of the matrix. The revised matrix **P** prime is shown below

$$\bar{\mathbf{P}} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

To ensure that the PageRank vector exists, the chain has to be irreducible and stochastic, we therefore make one more adjustments shown in the matrix below

$$\bar{\bar{\mathbf{P}}} = \alpha\bar{\mathbf{P}} + (1-\alpha)\mathbf{e}\mathbf{e}^T/n = \begin{pmatrix} 1/60 & 7/15 & 7/15 & 1/60 & 1/60 & 1/60 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 19/60 & 19/60 & 1/60 & 1/60 & 19/60 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 7/15 & 7/15 \\ 1/60 & 1/60 & 1/60 & 7/15 & 1/60 & 7/15 \\ 1/60 & 1/60 & 1/60 & 11/12 & 1/60 & 1/60 \end{pmatrix}$$

where $0 \leq \propto \leq 1$ and **E** = 1/n $\mathbf{e}^T$. It is now guaranteed that the application of the power method on the above matrix converges to a stationery PageRank vector π^T. A representation of the recursive PageRank formula is shown below

PR(A) = (1-d) + d (PR(T1)/C(T1) + ... + +PR(Tn)/C(Tn))

Where PR(A) is the PageRank of page A, PR(Ti) is the PageRank of pages Ti which link to page A, C(Ti) is the number of outbound links on page Ti and d is a damping factor which can be set between 0 and 1

2) HITS

Kleinberg (1999) developed the HITS algorithm; it later became part of the CLEVER Searching project at IBM Almaden Research Center. Extensions of the HITS ranking algorithm are currently in use by search engines such as "Ask Jeeves", which has acquired another search engine Teoma. Central to the idea of the HITS algorithm is the notion that a web page can assume one of two purposes, that is to make available information to a certain topic or link to other pages providing information on a topic, consequently, classifying web pages into hub or authorities (Kleinberg 1999). A web page is an authority on a specified subject if it provides authentic information on the said subject, and a web page is a hub if it provides links to pages (authority) that provide authentic information on a certain subject.

According to Kleinberg (1999) and Farhat et al (2006), HITS is typically applied to a subgraph of 1000-5000 nodes, which is constructed based on the search query terms corresponding to the ones found on the returned pages from a specified search. HITS exploit the hyperlinked nature of the web. It separates search topics into authorities of different bases; putting some pages in authoritative pool while others are placed in the hubs. The classification forms a link structure, which can be associated with Eigenvectors of certain matrices of the link graph, and this set up provides enough information for the study of link graph analysis (Herbach, 2001)

Supposing our sub graph is S, to construct the rankings, iteratively for each page p, we assign a non-negative authority weight $x^{(p)}$ and a anon-negative hub weight $y^{(p)}$. The invariant is maintained by normalizing each page so their squares sums to 1:

$$\sum_{p \in S} (x^{\langle p \rangle})^2 = 1 \text{ and } \sum_{p \in S} (y^{\langle p \rangle})^2 = 1$$

Therefore pages with larger x and y values are considered as better hubs and authorities. To show the mutually reinforcing relationships between hubs and authorities, we say that if p points to many pages with large x-values then it automatically has a large y value.

Conversely, if many pages with large y-values point to p, it should automatically have a large x-value. This gives rise to the mutually reinforcing rules, For the weights $\{x^{(p)}\}$, $\{y^{(p)}\}$,the mutual reinforcing events are given as follows:

The hubs update rule

$$y^{\langle p\rangle} \longleftarrow \sum_{q:(p,q)\in E} x^{\langle q\rangle} \; x^{\langle p\rangle} \longleftarrow \sum_{q:(q,p)\in E} y^{\langle p\rangle}$$

The authority update rule

$$y^{\langle p\rangle} \longleftarrow \sum_{q:(p,q)\in E} x^{\langle q\rangle}$$

To attain a balanced ranking corresponding to the values of the weights, we can apply the hubs and authority update rule repetitively until a fixed point is attained.

3) SALSA

SALSA is an improvement on HITS developed by Lempel and Moran (2000) by making adjustments to the work of Kleinberg (1999) by replacing the Mutual Reinforcement approach in HITS, and termed it the stochastic approach to which basically making the coupling between the hub pages and authorities pages less tight (Lempel and Moran, 2001). Assuming there exists a bipartite graph G with two pars; one forming the hubs and the other authorities. If there is a path between hub r and authority s means that there is a link between r to s. Therefore the dominant topic of pages in G should be traversable from many pages of G i.e there is high visibility. By looking closely at specified random walks in G, with the assumption that highly visible pages would be visited often more often than the other pages, it is clear that both Kleinberg's Mutual Reinforcement Approach and the stochastic approach used the same meta algorithmic solutions to different inputs of the graph (Lempel and Moran, 2000).

Furthermore, SALSA is less likely to face the topological phenomena known as the Tightly Knit Community (TKC) Effect, which in some cases hinders the effectiveness of the Mutual Reinforcement approach of HITS to identify useful authorities, therefore, justifying the advantage of SALSA over HITS as SALSA can identify meaningful authorities where HITS faces the TKC effect. The Stochastic approach is calculated by examining the weighted in/out degrees of the pages in G. Consequently, this proves that SALSA used less computational resources than the Mutual Reinforcement approach (Lempel and Moran, 2001)

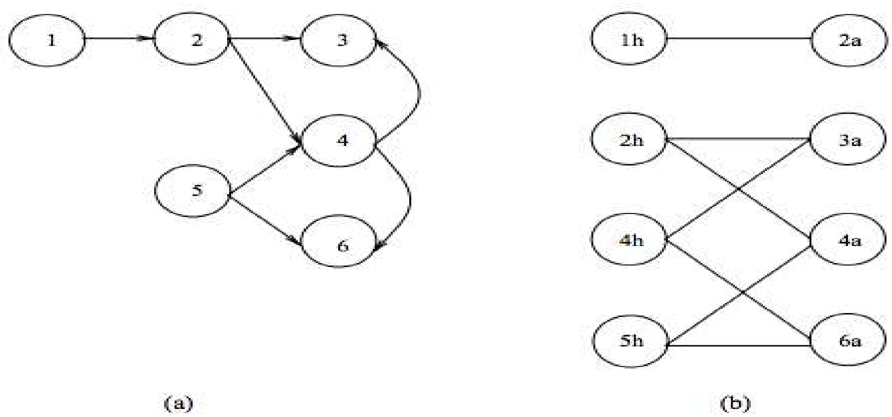

Fig 2: Transforming (a) the collect Z into a bipartite graph Ğ (b) (Lempel and Moran, 2011).

Building a bipartite undirected graph $\tilde{G}$ = (V_h, V_a E) from the above model adapted from Lempel and Moran (2001).

V_h = { S_h | s ∈ Z and out-degree(s)>0} (the hub side of $\tilde{G}$)

V_a = { S_a | s ∈ Z and in –degree(s) >0} (the authority side of $\tilde{G}$)

E = { (S_h, ra) | S → r in Z}

All non-isolated pages s ∈ Z are represented in $\tilde{G}$ by either S_h or S_a or both. Each link on the web) | s → r is represented by an undirected edge connecting Sh and r_a. From our model of the bipartite graph above, we will conduct two different random walks, alternating from one node on either side of the graph, by traversing paths of two $\tilde{G}$-edges in each step. Any walk is restricted to one of the graph's sides, and the two different walks will therefore start from different sides of $\tilde{G}$ because each edge crossed the side of $\tilde{G}$. It is pertinent to note that all paths of length 2 in Z represents a traversal one web link in the proper direction and a retreat when crossing from the other direction. (Lempel and Moran, 2001). If the hubs and authorities of a specified topic t should be highly visible in $\tilde{G}$, it suggests that the authorities of t are the most visited for every random walk on Va. Also, the hubs of t are the nodes visited for every random walk on V_h. The two walks on V_a and V_h will correspond a Markov chain each, which is the chain of visits to the authority part of $\tilde{G}$ and the chain of visits to the hub part of $\tilde{G}$. When these two chains are analyzed separately, it generates an authority score and a hub score, therefore the stochastic transition matrix of the Markov chains are shown next.

The hub Matrix H,

$$h_{i,j} = \sum_{\{k|(i_h,k_a),(j_h,k_a)\in\tilde{G}\}} \frac{1}{\deg(i_h)} \cdot \frac{1}{\deg(k_a)}$$

The authority matrix A are defined as follows

$$a_{i,j} = \sum_{\{k|(k_h,i_a),(k_h,j_a)\in\tilde{G}\}} \frac{1}{\deg(i_a)}.\frac{1}{\deg(k_a)}$$

A positive transition probability matrix $\bar{a}_{i,j}$ >0 means that a definite k points to both pages i and j and hence page j is reachable from page i by two steps by retracting along the link k → i and following the link k →j. Assuming that $\tilde{G}$ is connected then the stochastic matrices of A and H become irreducible, even if they are not connected to each can be treated differently with the same technique as the one connected suggesting that there is no limitation when A and H are not connected. It is important to note that both A and H are primitive, and the adjacency matrix of the support graph of A is symmetric (Lempel and Moran 2000).

4. CONCLUSION

Search engines have been in existence since the early 90s but the introduction of PageRank by google in the late 90s is what transformed the way search engines work. This paper has shown that search engines use a two-pronged approach to take advantage of the link structure of the web while finding a workaround to the large data size. The paper also showed that PageRank, HITS and SALSA can efficiently rank web pages to some extent, though SALSA is an improvement on HITS.

REFERENCES

1. Brin, S. and Page, L. (1998) 'The anatomy of a large-scale hypertextual Web [search engine' Computer networks and ISDN systems, 30 (1), pp. 107--117.
2. Ding, C., He, X., Husb, S, P., Zha, H. and Simon, H. D. (2002) 'PageRank, HITS and a unified framework for link analysis' pp. 353--354.Farahat, A., Lofaro, T., Miller, J. C., Rae, G. and Ward, L. A. (2006) 'Authority rankings from HITS, PageRank, and SALSA: Existence, uniqueness, and effect of initialization' SIAM Journal on Scientific Computing, 27 (4), pp. 1181--1201.
3. Haveliwala, T., Kamvar, S. and Jeh, G. (2003) 'An analytical comparison of approaches to personalizing PageRank' Stanford.
4. Herbach, J. D. (2001) 'Improving authoritative sources in a hyperlinked environment via similarity weighting' BSE Thesis, Dept. of Computer Science, Princeton University.
5. Kitchenham, B. A, Budgen, D., and Brereton, P. (2015). ‘Evidence-Based Software Engineering and Systematic Reviews’. Chapman & Hall/CRC.
6. Kleinberg, J. M. (1999) 'Authoritative sources in a hyperlinked environment' Journal of the ACM (JACM), 46 (5), pp. 604--632.
7. Langville, A. N. and Meyer, C. D. (2004) 'Deeper inside pagerank' Internet Mathematics, 1 (3), pp. 335--380.
8. Lempel, R. and Moran, S. (2001) 'SALSA: the stochastic approach for link-structure analysis' ACM Transactions on Information Systems (TOIS), 19 (2), pp. 131--160.
9. Lempel, R. and Moran, S. (2000) 'The stochastic approach for link-structure analysis (SALSA) and the TKC effect' Computer Networks, 33 (1), pp. 387--401.
10. Seymour, T., Frantsvog, D. and Kumar, S. (2011) 'History Of Search Engines.' International Journal of Management \& Information Systems, 15 (4).