International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 22 (2017) pp. 12305-12314

Efficient Method of Retrieving Digital Library Search Results using Clustering and Time Based Ranking

[1]Sumita Gupta, Neelam Duhan[2] and Poonam Bansal[3]

[1,2] *YMCA University of Science & Technology, Faridabad, Haryana 121006, India.*
[3]Surajmal Institute of Technology, Delhi, India

[1]Orcid:0000-0001-5947-9659

Abstract

Digital libraries are emerging as a significant source for serving the queries of researchers for relevant documents. With the growing digital content and the user's needs, the complexity of ranking mechanism utilized in digital libraries is increasing. Ranking plays an important role in digital libraries as it makes the user's search for scientific literature, research papers, or other academia based documents fruitful and avoids unnecessary navigation to find the desired content. Many ranking algorithms based on different parameters have already been proposed. The parameters like citations to a research paper, content of paper, impact factor of publication venue, age of the paper, bookmarks etc. are utilized for ranking the documents in the result list of the digital libraries. The existing ranking algorithms sometimes provide irrelevant results due to certain shortcomings, which indicate a scope for further improvement in ranking mechanism. In this paper an optimized ranking algorithm is proposed that carries out static as well as dynamic ranking to rank the documents in digital libraries. The proposed algorithm takes the link structure of the digital documents into consideration i.e. citations, bookmarks of the paper, paper age, user's feedback and clustering process for displaying efficient and relevant search result list. In this paper, an optimized approach is being proposed which provides sorted search result list in cluster foam against the user's query.

Index Terms— Document Clustering, Digital library, Web Mining, Page ranking, World Wide Web.

INTRODUCTION

World Wide Web (WWW) is composed of huge and massive volumes of information in the form of text, audio, video, images and metadata. It can be thought of as a large database possessing unstructured or semi-structured chunks of data [1]. For retrieving the more relevant results for users and researchers, digital libraries have been introduced. A digital library [2, 3] is an integrated collection of various services including catching, indexing, saving, finding, guarding and extracting digital content or information. It enables the user to easily access huge quantity of available digital information on web. Today, digital libraries are being utilized for various communities and in variety of different fields like academic, science, culture, health, and many more. Thus, the introduction of digital libraries has made the creation, storing, sharing and retrieving of information attractive and easy for the web users.

The amount of digital content in digital libraries is rapidly growing which somewhere degrading the results of the ranking mechanism utilized by the digital library search engines. Thus, the existing ranking algorithms still have some limitations which needs to be improve by optimizing document ranking in digital libraries. In this paper, an approach is being proposed named Time based Ranking and Clustering which uses Web Content, Web Structure as well as Web Usage Mining to display an ordered the search result list in accordance with the user interest. This paper is structured as follows: Section II, discusses about the review of some existing ranking algorithms has been discussed. Section III introduces with an optimized ranking method with its architecture, illustration and snapshots of system implantation results. Section IV presents a comparison summary of the proposed algorithm with some of the existing ranking algorithms. Finally in Section V, conclusion is drawn.

RELATED WORK

This section highlights the various existing ranking algorithms used to rank digital documents in online digital libraries till now. As per literature survey, it is concluded that digital libraries use the different parameters in order to rank the search results. The document clustering being employed by the search systems is also discussed in this section.

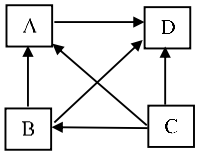
PAGERANK ALGORITHM

Page et al. [3] developed a ranking algorithm, named PageRank (PR) algorithm that considers the link structure of the documents into account for computing the ranking of the documents. This algorithm states if the paper has some important incoming links to it, then its outgoing links to other papers also become important. Thus, a paper gets a high rank if the sum of the ranks of its backlinks is high. The PageRank formula can be defined as:

$$PR(i) = (1-d) + d\sum_{j\in B(i)} \frac{PR(j)}{N_j} \qquad (1)$$

where *i* represents a paper, *B(i)* is the set of papers that point to *i*, *PR (i)* and *PR (j)* are rank scores of papers *i* and *j* respectively, N_j represents the total number of outgoing links of paper *j*, and *d* is the damping factor whose value ranges between 0-1 and is usually set to 0.85

Table 1: An Example of TDCC



Publication	Publication year
A	2011
B	2008
C	1998
D	1980

Figure 1: An Example of PageRank

ILLUSTRATION OF PAGERANK ALGORITHM

To illustrate the Pagerank, let us assume an example as given in Fig 1. The PageRanks for papers can be calculated by using (1):

$$PR(A) = (1-0.85) + 0.85\left[\frac{PR(B)}{2} + \frac{PR(C)}{3}\right] \qquad (1a)$$

$$PR(B) = (1-0.85) + 0.85\left[\frac{PR(C)}{3}\right] \qquad (1b)$$

$$PR(C) = (1-0.85) + 0.85[0] \qquad (1c)$$

$$PR(D) = (1-0.85) + 0.85\left[\frac{PR(A)}{1} + \frac{PR(B)}{2} + \frac{PR(C)}{3}\right] \quad 1d)$$

By calculating the above equations repeatedly until the page ranks get converged, the final page rank values for papers A, B, C and D are shown in the Table 2. The rank order for papers obtained is: PR (D)>PR (A) > PR (B) > PR (C).

Table 2: Final PageRank Values

PR(A)	PR(B)	PR(C)	PR(D)
1	1	1	1
0.85	0.430	0.15	1.09
0.375	0.192	0.15	0.592
0.274	0.192	0.15	0.507
0.274	0.192	0.15	0.507

TIME DEPENDENT CITATION COUNT

Marian et al. [4] proposed an extension to standard Citation Count method which is time-dependent approach, named as *Time Dependent Citation Count (TDCC)*. This method considers the year of publication of the citation and link structure of pages/papers while determining the importance of a document or publication. The method uses the time decay factor for knowing the freshness of the paper or citation in the citation graph.

The TDCC formula can be defined as:

$$Weight_i = e^{-w(t_p - t_i)} \qquad (2)$$

Where $Weight_i$ denotes the weight of paper *i*, t_i denotes the year in which publication *i* is published, t_p represents the present time (i.e. year), and *w* denotes the time decay factor whose value lies between 0-1 (w ϵ (0, 1]).

Illustration of TDCC Algorithm: To explain the working of TDCC, let us again refer to Fig 1 and Table 1. By using (2) weight scores of publications can be calculated as:

$$\mathrm{Wt}_A = e^{-w(2017-2008)} + e^{-w(2017-1998)} = e^{-w(9)} + e^{-w(19)} \qquad (2a)$$

$$\mathrm{Wt}_B = e^{-w(2017-19988)} = e^{-w(19)} \qquad (2b)$$

$$\mathrm{Wt}_C = 0 \qquad (2c)$$

$$\mathrm{Wt}_D = e^{-w(2017-2011)} + e^{-w(2017-2008)} + e^{-w(2017-1998)}$$
$$= e^{-w(6)} + e^{-w(9)} + e^{-w(19)} \qquad (2d)$$

where *w* is time decay factor. Let us take the threshold age = 10 years i.e. *w*=0 for the publications with the ages less than 10 years (considered as new publications) and *w*=1 for publications with ages more than 10 years (considered as old publications). By calculating the above equations, the rank score of publications become:

TDCC (A) = 1.000023, TDCC (B) = 0.000023, TDCC (C) =0

TDCC (D) = 2.000023

Here TDCC (D)>TDCC (A)> TDCC (B) > TDCC (C).

After doing the literature analysis of ranking algorithm, it may be noted that existing ranking algorithms have some limitations as they have returned a huge search result list to user irrespective of user's need and desire. Thus, these methods are not able to provide relevant results and satisfy the user's need in the concise manner. The returned result list of documents need to be arranged or ordered in a more user friendly manner. The paper also pays attention on some other methods or approaches like clustering that is used to represent the returned document list as per the user needs.

DOCUMENT CLUSTERING

Clustering [5] divides a set of objects into groups such that the objects in the same group are similar to each other. In the context of web document clustering [6], objects are replaced by documents and are grouped together based upon some measure like content similarity or link structure. As discussed earlier, most of the digital library search engines display a large and unmanageable list of documents against the user query. In order to find the relevant and desired documents from such a large list is usually tedious. To overcome this problem, the search engines can group or cluster a set of returned documents having same semantically meaning or belongs to same category. Document clustering may be based on content only, link only and may be based on both content and links. Popular clustering techniques like k-Nearest Neighbor [7] can be used for document clustering which categorizes the documents by comparing the category frequencies of the k-nearest neighbors. The Euclidean distance or the angle between the feature vectors is computed as a similarity measure between documents, but this algorithm is sometimes biased by the value of k i.e. number of clusters. Hierarchical clustering [8] can also be used for cluster analysis, but this is a greedy algorithm means optimized the result based on the currently available results or data. Thus, this method does not necessarily guarantee the best partition at a distant step.

By going through the available literature, few shortcoming in the in the existing search result organization techniques which needs a scope to improve are highlights as follows:

- First, complete processing is done on the fly (i.e. at run time) which results in degrading the system performance.
- Second, there is no method exist which considers both the link score and relevance score implied by the user surfing pattern for organizing the search results.
- Finally, a huge amount of search results in an order format is returned by most of the existing approaches, whereas few of them are generally accessed by the users.

Therefore, in order to overcome these shortcomings, an efficient clustering and ranking based approach has been proposed which provides an easy access to search result list organization against the query.

PROPOSED PAGE RANKING ALGORITHM

An efficient ranking algorithm is proposed which takes as input both the bookmarks of the papers and citations to the papers. Bookmarks are set of keywords that identify the document or some part of the document. For instance, the title, headings and the sub-headings are by default the bookmarks of the research document. This method displays the search result list as hierarchy of clusters relevant to user query. Moreover, the publications within the each cluster can be sorted or ordered as per their relevancy. This type of search result organization helps the user to limit his search by only going through the cluster having high query-cluster similarity score. In this algorithm, the relevancy score between the paper-paper and query-cluster is computed by considering the bookmarks of the publications only instead of taking the whole content of publication.

The proposed architecture (as shown in Fig. 2) of digital library search result optimization is consist of following functional modules: *Upload Module, Data Processing Module* and *Query Processing Module*. In this system, we assume, there are two types of user:

1. Administrator: -who can upload the papers in the database, and
2. End Users- who can only search the database for papers as per their interest.

When Administrator wants to upload any paper, then he/she will upload the paper into database through upload module. Data processing module takes the uploaded papers as an input

and computes the static ranking at the backend. When an end user hits a query in the form of query keywords through search engine interface, then query processing module processes the query and extracts the relevant results from the data processing module. These results are returned to the user.

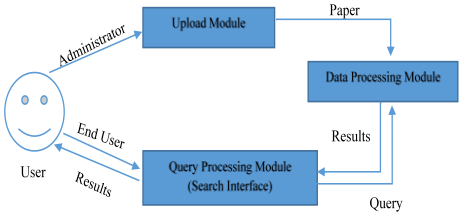


Fig 2 Architecture of proposed algorithm

The detailed overview of these modules is described below.

UPLOAD MODULE

When a user (only administrator) selects the option to upload a paper through *upload interface* (as shown in Fig 3), then the system first of all stores the paper in *Content Store*. In this

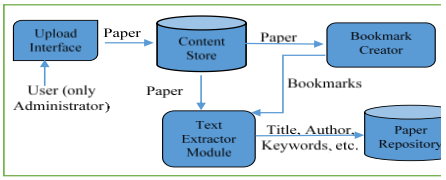


Fig 3 Upload Module

architecture, *Bookmark Creator* module becomes functional only when the newly uploaded paper does not contain bookmarks. This module creates the bookmarks of the newly uploaded paper and forwarded to Text Extractor Module which extracts the keywords from bookmarks. By using the bookmarks reading, the overall cost of the system is reduced in terms of space and time complexity. The Text Extractor module also extracts the important text (i.e. tittle, keywords, bookmarks, synonyms, authors name, references etc.) from the paper stored in content store. This extracted text is stored in a database called Paper Repository.

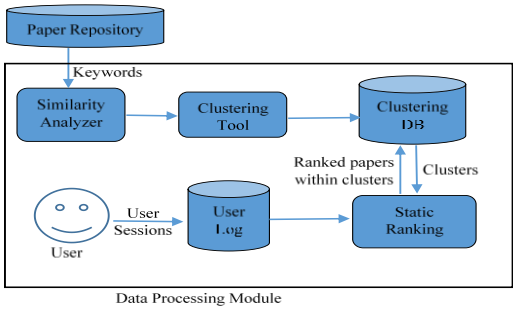


Fig. 4 Data Processing Module

DATA PROCESSING MODULE

This module (as shown in Fig. 4) contains four components: *Similarity Analyzer, Clustering Tool (Generator), User Logs*, and *Static Ranking* which are described below in detail.

A. *Similarity Analyzer*

Similarity between the publications means: which keywords or terms are occurred in the document, location or position of their occurrence and frequency of their occurrence in the documents? There are many methods to calculate the similarity between two publications, but here the proposed system considers the weight of the terms or keywords present in the document.

Similarity between the publication P and publication Q can be measured by computing similarity value [9] which is denoted by sim (P, Q). Generally, this value is ranging between 0 and 1. Cosine similarity measure is used to compute the similarity value between the two publications P and Q as shown below:

$$\mathrm{Sim}(P,Q) = \cos\theta = \frac{\mathrm{P.Q}}{\|\mathrm{P}\|\|\mathrm{Q}\|} = \frac{\sum_{i=1}^{n} W_{p,i} \times W_{q,i}}{\sqrt{\sum_{i=1}^{n} W_{p,i}^2} \times \sqrt{\sum_{i=1}^{n} W_{q,i}^2}} \quad (3)$$

where $W_{p,i}$ and $W_{q,i}$ denotes the weight of term t_i in the publication p and paper q respectively. These weights can be computed by calculating the frequency of occurrence of term t_i in P and Q.

B. *Clustering Tool (Generator)*

This component of the digital library search engine divides the data into groups of the similar publications. Each group, called cluster, contains documents that are more similar to each other than to those in other cluster. The clustering of publication is done based on the similarity values of the publications.

The algorithm works as follow: initially, all papers are assumed to be individual or not belong to any cluster. Each individual paper is examined against all other papers (whether classified or unclassified) by using (3). If the similarity score between the publications comes out to be higher as compared to the pre-specified threshold value (ƭ), then the papers are put into the same cluster or group. This process is repeated until all publications put into any one of the clusters. Finally, the returned clusters are stored in the Paper Cluster Database.

C. User logs

User Log is maintained that stores every user's session and is utilized to gain the number of downloads of every paper.

D. *Static Ranking*

The proposed ranking named time based ranking and clustering approach considers the three parameters named as *Download Score, Paper Posted Time (PPT)* and *Pagerank* for computing the static rank score of each paper in the cluster. These parameters are described as:

Download Score

This parameter extracts the no. of downloads of any paper (from user logs) to compute the download score for each paper in the cluster. Download score of paper P is calculated by using the equation (4) as shown below:

$$Download\ Score(P) = \frac{Numberof\ Downloads(P)}{Maximum\ Downloads} \quad (4)$$

Paper Posted Time (PPT)

Paper posted time is computed by using *Time Dependent Citation Count Algorithm (TDCC)* [4]. It utilizes the citation graph of nodes interconnected with each other through edges. Each node represents a research paper and an edge from one node A to node B represents a citation from paper A to paper B. Also, this method uses a time-decay factor which is applied to the citation counts to determine the weight of each node in the citation graph. The weight of paper i is computed by using the equation (8) as:

$$Weight_i = e^{-w(t_p - t_i)} \quad (5)$$

where t_p represents the current time i.e year, t_i denotes the year in which the paper i is published and w denotes the time-decay factor whose value lies between 0 and 1. For the paper age selected say 6 years, w=0 means that the paper is new and whereas w=1 means paper is old publication.

Page Rank

Page rank of the paper is calculated by using the *PageRank Algorithm* [3]. This method computes the rank of a paper by considering the number of citations (i.e backlinks) of the paper as described above in section II.

Thus, for each paper in a cluster, a static rank score is computed by using the equation (6) and is stored in the clustering database.

$$Static\ Weight = Download\ Score + PPT + PageRank \quad (6)$$

The papers within each cluster are rearranged on the basis of this static weight.

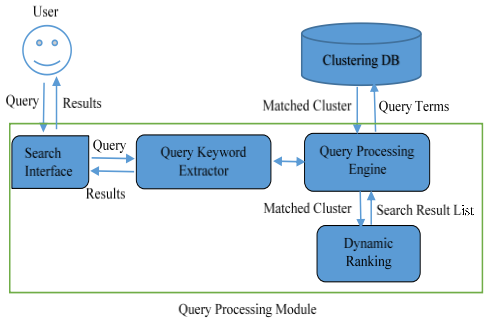


Fig 5 Query Processing Module

QUERY PROCESSING MODULE

When a user fires a query in term of query keywords on the *Search Engine Interface* (as shown in Fig 5), the Q*uery Keyword Extractor* extracts the query keywords. The *Query Processing Engine* matches the query terms with the *Clustering DB* and returns the matched cluster of publications in response to the query. The *Dynamic Ranking* component works on the fly and considers the matched cluster retrieved by query processing engine as an input. It further improves the rank of papers based on the score assigned to each paper according to the similarity between the query and paper. Thus, the similarity between the query q and paper d is calculated by using (7) [9]:

$$\text{sim}(q, d) = \frac{\sum W_{q,j} \times W_{d,j}}{\sqrt{\sum W^2_{q,j}} \times \sqrt{\sum W^2_{d,j}}} \quad (7)$$

where $W_{q,j}$ and $W_{d,j}$ denotes the weight of term t_j in the query q and paper d respectively. These weights can be computed by calculating the frequency of occurrence of term t_j in q and d.

It may be noted that the papers in the matched cluster are ranked as per the new rank as shown below:

$$Rank(p) = Static\ Weight(p) + Sim(p, q) \quad (8)$$

This ranking method takes importance of the paper, the relevance of citations and user intensions too into account for computing the ranking of search results. Now the user is returned with a cluster of ranked papers within it. Now it's up to the user that which paper he opens based on the rank values.

Illustration of Proposed Algorithm: An example is taken to explain the ranking mechanism of the proposed algorithm. The existing papers in the database are shown in Table 3.

Firstly, bookmarks from each paper in the database are extracted which are utilized for determining the frequently occurring keywords for each paper. Table 4 lists the frequently occurring keywords of the papers in the database.

Similarity Analyzer: Now, the similarity analyzer will compute a similarity score between the already existing papers in the

Table 3: Paper Repository

S.No	Paper Title	Paper Year	Down - loads
A	Page Ranking Algorithms for Web Mining	2011	9
B	Web Crawler Architecture	2000	9
C	How search engines work and a web crawler application	2011	8
D	Comparative study of Page Ranking Algorithms for Web Mining	2013	7
E	Mercator: A scalable, extensible Web crawler	2013	7
F	Web Crawler: Extracting the Web Data	2013	8
G	A Survey- Link Algorithm for Web Mining	2011	7
H	Analysis of Various Web Page Ranking Algorithms in Web Structure	2014	10

	Mining		
I	Application of Page Ranking Algorithm in Web Mining	2012	8
J	Web Mining Research: A Survey	2000	8

database by using equation (3). The similarity scores computed after the comparison of the research papers in the database are presented using a matrix namely similarity matrix as shown in Table 5.

Clustering: Now, the clusters of the research papers existing in the database are formed using the similarity matrix with the threshold value 0.2 (assumed) and saved for future use. On the basis of the similarity matrix three clusters are formed as shown in Table 6.

Table 4: Most frequent keywords in each paper in the database

S.No	Keywords
A	Web, Mining, Page, Algorithms, Ranking, Categories, Content, Structure, Usage, Link
B	Web, Crawler, Architecture, Historical, Background, Foundation, Key, Application, Future, Directions
C	Web, Search, Indexing, engines, crawler, crawling, application, content, work, popular
D	Ranking, Algorithms, Page, Comparative, study, Web, Mining, Text, Link, Analysis
E	Crawler, scalable, Web, Mercator, extensible, Architecture, Extensibility, traps, hazards, Results
F	Web, Crawler, Crawling, Extracting, Data, Literature, Survey, Architecture, Types, Algorithms
G	Web, Mining, Page, Rank, Weighted, Content, Link, Algorithm, Algorithms, Survey
H	Web, Page, Ranking, Algorithms, Analysis, various, Structure, mining, Comparison
I	Algorithm, Page, Ranking, Web, Mining, Application, Methodologies, Weighted, Rank, HITS
J	Mining, Web, View, Research, Survey, Overview, Categories, Agent, Paradigm, Content

Static Ranking: Next, static ranking mechanism is performed for computing the weight for each paper within a cluster. The static ranking considers number of downloads, TDCC and page rank of the paper. The number of download of each paper is assumed in this example which is utilized to compute an average download score for every paper.

Download Score*:* To calculate the number of download score of paper, extract the number of downloads of each paper from Table 3. The maximum number of downloads is considered to be 10 and calculations are done by using (4) as shown in Table 6.

Output of PPT*:* To calculate the paper posted time of the papers, extract the publish year of all the papers. Assume *w* i.e. time decay factor = 6 years. Then, apply the formula of time dependent citation count algorithm by using (5). After solving above equations, paper posted time computed for paper in each cluster is listed in Table 6.

Output of Pagerank*:* The page rank of the papers in the database is computed by using the equation (1). In this algorithm, d is set to 0.85 and calculations are done as shown in Table 6.

Static Weight: The computed static weight by using (6) for each paper in each cluster is listed in Table 6. The total weight of each paper within a cluster is obtained by adding all the three

The total weight of each paper within a cluster is obtained by adding all the three parameters. Then, each cluster is rearranged according to the computed weight of the papers.

Table 5: Similarity Matrix

	A	B	C	D	E	F	G	H	I	J
A	1									
B	0.204	1								
C	0.376	0.323	1							
D	0.488	0.052	0.091	1						
E	0.233	0.391	0.424	0.059	1					
F	0.519	0.438	0.635	0.188	0.656	1				
G	0.830	0.163	0.325	0.401	0.185	0.422	1			
H	0.744	0.186	0.324	0.751	0.211	0.494	0.607	1		
I	0.455	0.124	0.165	0.393	0.094	0.198	0.576	0.463	1	
J	0.886	0.198	0.360	0.224	0.225	0.486	0.26	0.510	0.353	1

Table 6: Final Rank Values

Cluster No.	S.No	Paper Title	Download Score	Page Rank	PPT	Static Weight	Sim (q,c)	Dynamic Rank	Rank
I	A	Page Ranking Algorithms for Web Mining	0.9	0.1925	0.0371	1.1296	0.566	0.752	1.8816
	D	Comparative study of Page Ranking Algorithms for Web Mining	0.7	0.3491	0.0470	1.0961		0.223	1.3191
	G	A Survey- Link Algorithm for Web Mining	0.7	0.1909	0.0470	0.9379		0.657	1.5949
	H	Analysis of Various Web Page Ranking Algorithms in Web Structure Mining	1	0.5647	0.0223	1.587		0.748	2.335
	I	Application of Page Ranking Algorithm in Web Mining	0.8	0.2720	0.0272	1.0992		0.482	1.5812
	J	Web Mining Research: A Survey	0.8	0.15	0	0.95		0.549	1.499
II	B	Web Crawler Architecture	0.9	0.15	0	1.05	0.213		
	C	How search engines work and a web crawler application	0.8	0.2562	0.0743	1.130			
	E	Mercator: A scalable, extensible Web crawler	0.7	0.5020	0.0545	1.2565			
	F	Web Crawler: Extracting the Web Data	0.8	0.6855	0.0148	1.5003			

Hence, the sequence of the research papers stored in the clusters formed is,

Cluster I: **H, A, I, D, J, G**

Cluster II: **F, E, C, B**

Table 7: Keywords attached to each cluster

Cluster No.	Keywords
I	web, mining, rank, algorithms, page, ranking, structure, link, categories, content, weighted, algorithm
II	Web, crawler, architecture, application, crawling, historical, background, foundation, key, future, directions, search

After the retrieval of clusters, Clusters are saved in the cluster database along with most frequently occurred set of keywords as listed in Table 7.

Let the user query be Q, which the user submits to the search engine through query interface for retrieving the relevant documents.

Query Q: *Concept of page ranking algorithms in web mining*.

The query keyword extractor extracts the keywords from the user's query which are listed below,

Query Keywords: Concept, page, ranking, algorithms, web, mining.

Now, the query processing engine will match the above listed keywords with the cluster keywords mentioned in Table 7 so as to select the appropriate cluster for serving the user's query. The similarity score between the query and the cluster keywords is computed using the equation (7) as show in Table 6.

Clearly, it can be seen that the cluster I is the suitable cluster for forming the result set of the query fired. Now, the papers in the matched cluster will be rearranged according to rank computed by using (8). The papers in the cluster I will be re-ordered according to the total weight computed and will be displayed to the user as search result set. The final result set provided to the user is shown in Table 8.

Table 8: Final result set against the user's query

S.No	Paper Title
H	Analysis of Various Web Page Ranking Algorithms in Web Structure Mining
A	Page Ranking Algorithms for Web Mining
G	A Survey- Link Algorithm for Web Mining
I	Application of Page Ranking Algorithm in Web Mining
J	Web Mining Research: A Survey
D	Comparative study of Page Ranking Algorithms for Web Mining

SNAPSHOTS OF IMPLEMENTED SYSTEM

This section presents the implementation details and the experimental results that have been performed. The proposed ranking system is implemented using the platform Java JDK 6.0 and software's mySql 5.6, Apache PDFBox 1.8.9, WordNet Version 3.0 and MS-Access. Experiments for clustering the research papers is performed on Dual-Core Intel Pentium IV or higher Processor with 2.60GHz frequency and 4.00 GB RAM. NetBeans IDE is used for the implementation of the proposed system. For parsing the PDF file, PDFBox is used.

There are some screen shots given which will help user to show the results.

Figure 6: User Interface

Fig. 6 shows the interface of a Ranking System for online digital library. By authorizing the user (which is not shown here because of simplicity), the paper can be uploaded to content store. When the user hits the query (as shown in Fig 7), then first query keyword extractor extracts the keywords from query and processed the query.

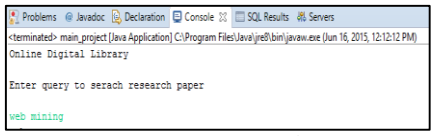


Figure 7: Query Interface

The query processing engine finds the relevant cluster (as shown in Fig 8) against the query by comparing the query keywords and cluster keywords. Once the relevant cluster is found, paper weights are updated (as shown in Fig 9) within that cluster according to the user query relevance with the papers by using dynamic ranking. After that, final search result list is returned and displayed to the user.

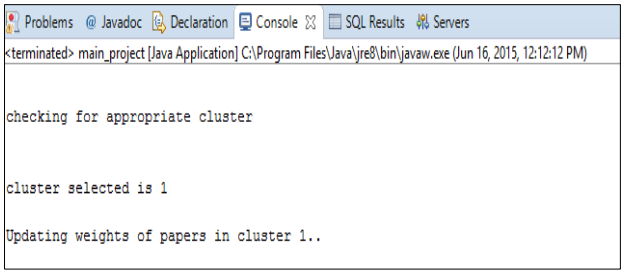


Figure 8: Result of Relevant Cluster

\

Figure 9: Final updated PageRank Values

COMPARISON

A critical look at the available literature concludes that each algorithm has some relative strengths and limitations. The proposed *Clustering and Time based Ranking (CTR)* method ranks the results in order to organize them in an efficient and easily accessible manner as compared to the returned search result list by PageRank (PR) and Time Dependent Citation Count (TDCC) algorithms. CTR algorithm considers combination of Web Content, Web Structure and Web Usage mining for ranking the more relevant results at the top of search result list as compared to PR and TDCC. The comparison of the three ranking algorithms PR, TDCC and CTR based on different parameters is shown in Table 9.

CONCLUSION

The existing ranking approaches possess few limitations due to which they sometimes fail to display effective results against the user's query. Since, the researchers depend on the digital libraries for retrieving the needful information content, therefore it is necessary to overcome these shortcomings. The paper presents an optimized ranking approach that enhances the ranking mechanism and provides better and relevant results than the existing algorithms. The existing algorithms are either based on content similarity or link structure. But, this proposed approach takes into account the link structure of the papers i.e. citations, bookmarks of the paper, paper age, user's feedback and clustering process for displaying efficient and relevant search result list. In this paper, a method is proposed which returns sorted search results list in cluster foam against the user's query.

Table 9: Comparison of PageRank, TDCC and Proposed Algorithm

Algorithms → / **Measures ↓**	**PageRank**	**TDCC**	**Proposed Algorithm (Clustering and Time based Ranking)**
Main Technique used	Web structure mining	Web structure mining,	Web Structure Mining, web content mining, web usage mining, Clustering
Description	Computes the score at indexing time. Papers are sorted according to importance of citing paper.	Papers are sorted based on age of the citations.	Results are ranked by taking into account the link structure as well as content similarity among the papers. It also involves clustering of papers for enhancing the results.
Input Parameters	Backlinks	Incoming links, Paper posted time	Bookmarks, query's content, paper posted time, number of downloads
Relevancy of papers	No	No	Yes
Quality of results	Low	Higher than PageRank	High
Nature of Rank	Less dynamic (rank changes with link structure)	Less dynamic (rank changes with weightage of year)	More dynamic (rank changes with downloads & structure of links)
Advantages	Traditional method. Computation of ranks with minimum efforts and less complexity.	Higher weightage is given to new citations as compared to old citations.	Clusters are formed based on the similarity and rank the papers within the cluster by using static and dynamic ranking score.
Limitations	Relevancy of papers is not considered while computing the rank.	Instead of considering Relevancy of citations, published year of the citation is considered.	More complexity in terms of time and space.

REFERENCES

[1]. Duhan, N., Sharma, A., and Bhatia K., 2009, "Page Ranking Algorithms: A Survey," IEEE International Conference on Advance Computing, pp. 2811-2818.

[2]. Krishnamurthy, M., 2008, "Open access, open source and digital libraries: A current trend in university libraries around the world. A General Review," Emerald Group Publishing Limited.

[3]. Gupta, S., Duhan, N., and Bansal, P., 2013, "A comparative study of page ranking algorithms for online digital library", International Journal of Scientific & Engineering Research, 4(4), pp. 1225- 1233.

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

[5]. Marian, L., LeMeur, J., Rajman, M, and Vesely,M., 2010, "Citation Graph Based Ranking in Invenio," in ECDL 2010: Research and Advanced Technology for Digital Libraries, pp. 236-247.

[6]. Han, J., and Kamber, M., 2006, "Data Mining: Concepts and Techniques," Academic Press, San Francisco, London, Morgan Kaufmann Publishers.

[7]. Toda, H., and Kataoka, R., 2005, "A search result clustering method using informatively named entities," WIDM'05 7th annual ACM international workshop on Web information and data management, pp. 81-86.

[8]. Dasarthy; B.V., 1991, "Nearest Neighbour (NN) Norms: NN pattern classification techniques," California, McGraw Hill Computer Science Series, IEEE Computer Society Press.

[9]. Lawrie, D.J., and Croft, W.B., 2003, "Generating hierarchical summaries for web searches," Proc. of 26th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 457-458.

[10]. Gomes, M., Júnior, C., and Gong, Z., 2005, "Web Structure Mining: An introduction," Proc. Of IEEE International Conference on Information Acquisition, China.

[11]. Chaudhari, H.C., Wagh, K.P., and Chatur, P.N., 2015, "Search Engine Results Clustering Using TF-IDF Based Apriori Approach," in International Journal of Engineering & Computer Science, 4(5), pp. 11956-11961.

[12]. Chen, J., Zaiane O.R., and Goebel, R., 2008, "An Unsupervised Approach to Cluster Web Search Results based on Word Sense Communities," 2008 IEEE/WIC/ACM, International Conference on Web Intelligence and Intelligent Agent Technology.

[13]. Duhan, N., and Sharma, A., 2010, "A Novel Approach for Organizing Web Search Results using Ranking and Clustering," International Journal of Computer Applications, 5 (10), pp. 1-9.

[14]. Roul, R.K., Devanand, O.R., and Sahay, S.K., 2014, "Web Document Clustering and Ranking using Tf-Idf based Apriori Approach," in IJCA Proceedings on International Conference on Advances in Computer Engineering and Applications, pp. 34–39.