

# A Novel Approach for Document Ranking in Digital Libraries using Extractive Summarization

Shikha Singla  
M-Tech scholar, YMCAUST  
Sector 6, faridabad

Neelam Duhan, Ph.D  
Associate Professor,  
YMCAUST  
Sector 6, faridabad

Usha Kalkal  
M-Tech scholar, YMCAUST  
Sector 6, faridabad

## ABSTRACT

Digital libraries have become an important tool for searching the scientific literature. The growth of digital libraries is exponential based upon the fact that researchers are utilizing it more and more by the days. To make the search better, the content of digital libraries should be ranked properly. The rank of research paper depends upon many factors like citations to paper; content, authors and publications of the paper etc. Based upon these parameters, different ranking algorithms have been proposed till now. In this paper, a new algorithm C3 ranking is proposed, which considers two important factors i.e. citations to the paper and the relevancy of the content with the query.

**Keywords:** Digital libraries, C3 Ranking, Summary, Citation

## 1. INTRODUCTION

World Wide Web [1] is a collection of heterogeneous data like image, audio, video, text etc. As we can see, the size of WWW is increasing exponentially. Therefore, some information retrieval techniques are used by user to find and

filter the desired information. Commercial search engines like Google, Alta-Vista and Yahoo etc return a long list of results containing documents, news articles, tutorials, blogs, pdf documents etc. A user, with the intent of research, mainly looks for research papers. So, it was very turbulent for him to go through the search results in order to find relevant papers.

To overcome this problem, digital libraries came into picture. A digital library [2] is an integrated set of services for capturing, cataloging, storing, searching, protecting, and retrieving information, which provide coherent organization and convenient access to typically large amounts of digital information. Now days, the growth of digital libraries is exponential with respect to both the amount and richness of available digital content. Due to the availability of large amount of digital content, modern search engine technologies are now being introduced in digital libraries to retrieve the relevant content. The architecture of a typical digital library search system is shown in Fig.1. In this architecture, the main module is crawler which traverses the specified articles from the web and downloads them.

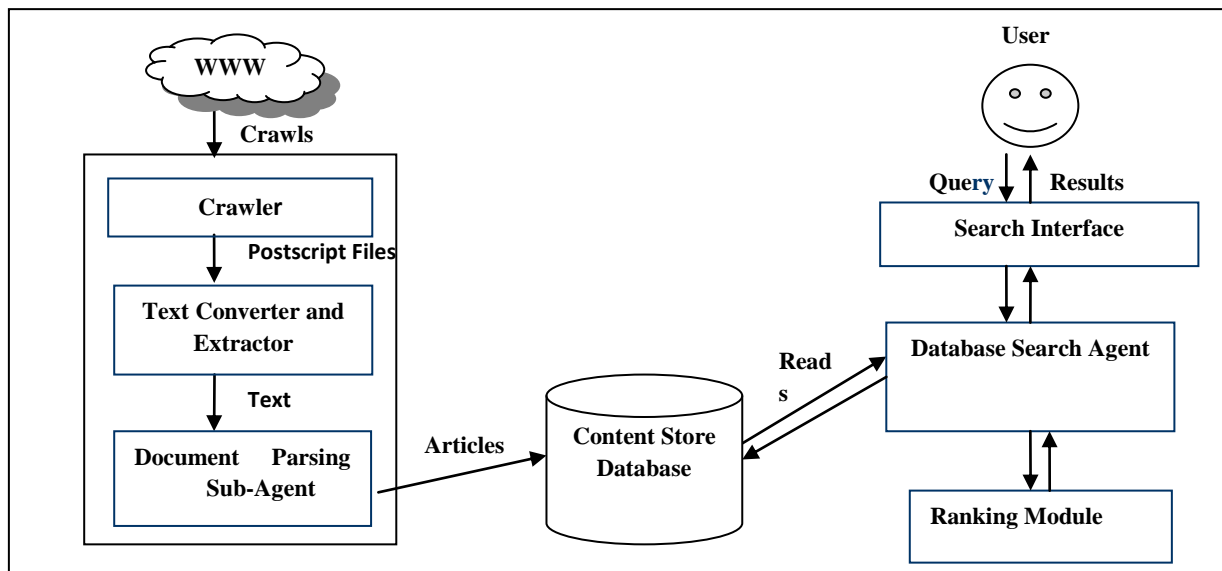


Fig 1: The Architecture of Digital Library Search Engine

Almost all the research articles are generally in the form of PDF and postscript files i.e. “.ps”, “.ps.Z”, or “.ps.gz” extension. So, after downloading the articles, they are converted into text files for extracting text from these articles. These downloaded files are passed through the document parsing sub agent which extracts the semantic features from the downloaded articles and places these parsed files into

database. When a user fires a query, the digital library search engine returns results containing a set of research papers or articles related to query. Database search agent retrieves the papers from content store database by taking the user query in proper syntax. Then, these related papers are ordered by ranking module and presented to the user in an ordered form.

This paper is structured as follows: First, an overview of related work is given including some prevalent ranking algorithms of typical digital libraries. Then, next section describes the Content Based Citation Count (C3) Ranking algorithm which has been proposed for ordering search results in digital libraries. After this, fourth section explains the working of proposed algorithm with the help of example. A comparison of proposed algorithm with previous algorithms is carried out in fifth section. In the conclusion section, our interpretation of the results is presented and an outlook towards further research is given.

## 2. RELATED WORK

This section discusses about various ranking algorithms for digital libraries proposed till now. Research papers have many features based on which different rankings could be performed. These features are citations to the publication, content, authors, publication year and journal of the publication etc. As per research, we concluded that different digital libraries rank their results on the basis of different factors like *Science Direct* ranks its results according to date and relevance; *IEEE Explore* offers a ranking on the basis of title, *ACM Digital Library* gives the choice to select the ranking based upon publication year, citation counts, alphabetically by title or journal and relevancy. All these digital libraries use different ranking algorithms to rank their papers. A brief description of various existing ranking algorithms [3,4] is given below:-

*Page Content Rank Algorithm:* - Jaroslav Pokorný et al.[5] gave a ranking method of page relevance ranking employing Web Content Mining(WCM) technique, called *Page Content Rank (PCR)*. This method combines a number of heuristics that seem to be important for analyzing the content of web pages. Here, page importance is determined on the basis of the importance of terms contained in the page; while the importance of a term is specified with respect to a given query  $q$ . PCR uses a neural network as its inner classification structure. In PCR, for a given query  $q$ , resultant papers are in turn classified according to their importance. Here a page is represented in a similar way as in the vector model and frequencies of terms in the page are used.

*Citation Count ranking algorithm:* One of the most frequent used ranking algorithm for measuring a scientist's reputation, named Citation Count was proposed by Joeran Beel et al. [6]. In this algorithm, the importance of the paper is based upon the number of citations to it. More the number of citations to the paper, higher would be its rank. This is the most commonly used ranking algorithm in digital libraries. Citation count ranking is defined as:

$$CC_i = |I_i| \quad (1)$$

Where  $CC_i$  is the citation count of the paper  $i$ ,  $|I_i|$  is the number of citations to paper  $i$ .

*Time Dependent Citation Count Ranking Algorithm:* There may be the case, the user need recent relevant papers instead of most cited papers. So, Ludmila Marian [7, 8] proposed an extended version of Citation Count Algorithm i.e. *Time Dependent Citation Count Ranking Algorithm*. In this algorithm, the time of the citation is also considered with the number of citations to rank the paper. It means that the importance of the paper depends upon the age of the paper. The paper is more important if the age lies within given time

interval. According to this algorithm, the weight of the paper is calculated by:

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

Where  $Weight_i$  represent the weight score of paper  $i$ ,  $t_i$  denotes the published year of paper  $i$ ,  $t_p$  denotes the present time (i.e. year), and  $w$  denotes the time decay parameter ( $w \in (0; 1]$ ), which quantifies the notions of 'new' and 'old' citations.

*PageRank Algorithm:* Sergey Brin and Larry Page [9,10] proposed ranking algorithm in which outgoing links from the paper are also considered along with the incoming links. The weightage of the incoming link is higher if the link is coming from an important paper. To calculate the rank of the paper by this algorithm, a formula is given below-

$$PR(u) = (1 - d) + d \sum_{v \in B(u)} \frac{PR(v)}{N_v} \quad (3)$$

Where  $u$  represents a paper,  $B(u)$  is the set of papers that point to  $u$ ,  $PR(u)$  and  $PR(v)$  are rank scores of papers  $u$  and  $v$  respectively,  $N_v$  denotes the number of outgoing links of paper  $v$ , and  $d$  is a normalization factor which lies between 0 and 1.

*Popularity Weighted Ranking algorithm:* Yang Sun and C. Lee Giles [11] proposed a new PageRank Algorithm with improved performance, named as *Popularity Weighted Ranking algorithm*. It came with the concept of popularity of the venue of publication. It means this algorithm considers the importance of the venue along with the weighted of incoming links to the paper.

The existing page ranking algorithms indicates the following limitations which need to be addressed while devising a novel page ranking method for digital libraries:-

- To rank the papers according to content, important keyword need to be extracted. Keyword extraction techniques are very time consuming as they need to scan full paper to extract the important keywords.
- It does not take into account the importance of citing paper i.e. citation from the reputed journal get the equal weightage as the citation from the poor journal.
- Importance of the paper/article to be ranked is totally ignored in this method. Importance of the paper is depends upon mainly three factors: (i) how many papers referred the paper/article, (ii) how many browsed the article, and (iii) in which journal or conference, the paper is published.

## 3. CONTENT BASED CITATION COUNT RANKING ALGORITHM

Content based Citation Count (C3) ranking algorithm is the proposed page ranking algorithm which employs summarization of research papers as well as citations to the publication as the input. This algorithm is a combination of two basic ranking methods i.e. Citation Count ranking and Content based ranking. Citation Count is a frequently used ranking algorithm for measuring a scientist's reputation. This method uses the citation graph of the web to determine the ranking of scientific work. This method states that if a publication has more number of citations to it, then publication becomes important. But some authors misuse it to have their publications highly ranked during search. To overcome this problem, the proposed algorithm uses the content of the paper which cited the publication along with the

number of citations. In this algorithm, the relevancy score between the publication and the paper which cited the publication is computed on the basis of its content. To check whether the papers are related or not, it uses the summaries instead of comparing the whole content of the papers.

The method works in following phases:

1. Calculation of summaries of papers
2. Similarity calculation of papers

### 3.1 Calculation of summaries of papers

To create a summary of the paper, important keywords of the papers are needed. To extract these keywords, system reads one paper at a time from the downloaded papers and extracts the keywords of the selected paper from the title, explicit keywords and titles of the papers in the references. It, in turn, sorts the extracted keywords on the basis of number of occurrences and selects top ten frequent keywords. After this phase, summarization is done on the basis of keywords. A

Research paper is mainly composed of these four parts or sections:-

- Abstract
- Introduction
- Related work
- Proposal or Implementation or Heading containing keywords from the title

The number of parts may increase or decrease depending upon the work described in the paper. *Abstract* is a short and the compendious idea of the whole research paper; *Introduction* gives the main concept of the paper, so it has the highest weightage in the summary; *Related work* contains work done in the field till the paper is published and in the end, paper has the *proposal*. So, based on the importance, this module selects 15, 35, 20 and 30 percent lines of total summary of these different sections respectively. Sentences are selected on the basis of occurrences of important keywords. Now, the algorithm in Fig 2 describes the creation of summary of the paper:-

**Algorithm 1-Summarization (Data,p)**      // *p* is a research paper or article.

```

{
1. keywords ← Key_Extract(paper)      //Keyword is a variable which contains keywords
                                   of the paper.
2. Data ← keywords
3. Summary ← Summarize(keywords, paper)      //summary is a data struture which
                                   contains summary of a paper.
4. Summary_data ← summary
}

```

**Fig 2: Algorithm for Summarization of Research paper**

This algorithm firstly, extracts the important keywords from the paper and stores them into the database. Then by using these keywords, creates a summary of the paper and stores in the summary database. The description of the various functions used in the algorithm is given below:

1. Key\_extract() :-  
Read the paper, and then extract keywords from title, explicit keywords and title of references. It further checks the frequency of the keywords and returns top ten high frequency keywords.
2. Summarize() :-  
This section selects 15, 35, 20 and 30 percent sentences of total summary respectively from Abstract, Introduction, Related work and Proposal or Implementation or Heading containing keywords from the title according to occurrence of keywords.

### 3.2 Similarity calculation of papers

After the summary creation, apply some similarity measures to compute the similarity between summaries of two papers. *Linguistic Measures* [12] are used for this purpose. These measures utilize linguistic knowledge such as semantic relations between words and their syntactic composition to determine the similarity of sentences. Three major linguistic approaches are used for this comparison, which can be described as follows:-

#### • Sentence Semantic Similarity Measures

First a semantic-vector approach[13] is used to measure sentence similarity. Sentences are transformed into feature vectors having individual words from the sentence pair as a feature set. Term weights are derived from the maximum semantic similarity score between words in the feature vector and words in the corresponding sentence. Here the semantic similarity of words within the same part-of-speech class is calculated. Finally, semantic similarity between the sentence pair is defined as a cosine similarity between semantic vectors of the two sentences. Given two sentences  $s1$  and  $s2$ , the sentence similarity calculation begins by finding the maximum word similarity score for each word in  $s1$  with words in the same part of speech class in  $s2$ . Then, apply the same procedure for each word in  $s2$  with words in the same part of speech class in  $s1$ . The derived word similarity scores are weighted with  $idf$  scores that belong to the corresponding word. The sentence similarity formulation is defined in (4).

$$Sim_{sem,IDF}(S1,S2) = \frac{1}{2} \left( \frac{\sum_{w \in s1} (maxsim(w,s2) \times idf(w))}{\sum_{w \in \{s1\}} idf(w)} + \frac{\sum_{w \in s2} (maxsim(w,s1) \times idf(w))}{\sum_{w \in \{s2\}} idf(w)} \right) \quad (4)$$

where  $maxSim(w,si)$  is the maximum semantic similarity score of  $w$  and words in  $si$  that belong to the same part-of-speech as  $w$  while  $idf(w)$  is an inverse document frequency of  $w$ . The reason for only computing the semantic similarity scores between words in the same part of speech class is that most

WordNet-based measures are unable to compute semantic similarity of cross-part-of-speech words.

- *Word Order Similarity*

Apart from lexical semantics, word composition also plays a role in sentence understanding. Basic syntactic information, such as word order, can provide useful information to distinguish the meaning of two sentences. This is particularly important in many similarity measures where a single word token was used as a basic lexical unit when computing the similarity of sentences. Without syntactic information, it is impossible to discriminate sentences that share the similar bag-of-word representations. For example, “the sales manager hits the office worker” and “the office manager hits the sale worker” will be judged as identical sentences because they have the same surface text. However, their meanings are very different. To utilize word order in similarity calculation, Li et al. [13] defines word order similarity measure as the normalized difference of word order between the two sentences. Word order vector  $r_i$  is a feature vector whose feature set comes from words that appear in a sentence pair. The index position of the words in the corresponding sentence are used as term weights for the given word features.

- *The Combined Semantic and Syntactic Measures*

Based on the notion that both semantic and syntactic information contribute to the understanding of a sentence, Li et al.[13] defined a sentence similarity measure as a linear combination of semantic vector similarity and word order similarity. The relative contribution of semantic and syntactic measures is controlled by a coefficient  $\phi$  or  $\psi$ . It has been empirically proved [14] that a sentence similarity measure performs the best when semantic measure is weighted more than syntactic measure. The relation between these semantic and syntactic measures is as follows:-

$$\text{Combined Similarity} = \phi (\text{Semantic measure}) + \psi (\text{Syntactic measure}) \quad (5)$$

Where  $\phi$  and  $\psi$  are the numerical constants with  $\phi + \psi = 1$ ,  $0 \leq \phi, \psi \leq 1$ .

The formula given below calculates the C3 rank of the paper:

$$C3\_Rank(p) = \frac{\sum_{j \in B(p)} \text{sim}(\text{sum}_p, \text{sum}_j)}{|B(p)|} \quad (6)$$

Where  $p$  is the paper or article for which rank is to be calculated,  $B(p)$  denotes the backlinks of the paper,  $\text{sum}_i$  is the summary of the paper  $i$  and  $\text{sim}(k, l)$  gives the similarity score between the summaries  $k$  and  $l$ . Now, the detailed description of C3 ranking is given by the algorithm presented in Fig 3:

```

Algorithm 2-C3_ranking(Data,Summary_data,p)    //p is a research paper or article.
//summary1, summary2 are variables which contain summary of the article.
{
1.  u ← Ref_extract(p,Data)    //u is data structure which contains papers having reference
                                to the paper p.
2.  sum=0    // sum is a variable which contains sum of all similarity factor.
3.  summary1 ← Summary(p,Summary_data)
4.  For(i=0;i<u.length;i++)
    {
5.    summary2 ← Summary(u[i],Summary_data)
6.    result←Compare(summary1, summary2) // result is a variable which contains similarity factor.
7.    sum=sum+result
    }
8.  rank=sum/u.length
9.  Return rank
}

```

**Fig 3: Algorithm to compute C3 Rank**

This ranking algorithm first extracts the papers having references to the publication from the database and stores in a data structure  $u$ . Then it extracts the summaries of the publication and papers contained in variable  $u$  from the summary database. Now, it computes the similarity factor between two papers by comparing their summaries. At last, calculates the rank of the publication by dividing the sum of similarity value between the publication and papers within  $u$  by total number of papers in  $u$ . The description of the various functions used in the algorithm is given below:

1. **Ref\_extract() :-**

This function extracts the papers having reference to the publication and stores in the database.

2. **Summary() :-**

This function extracts the summary of respective paper and stores into a data structure.

3. **Compare() :-**

This function compares the two summaries line by line and assigns a value according to the similarity between them by using Linguistic measures.

#### 4. EXAMPLE ILLUSTRATION

In this section, some examples are taken to explain the working of the C3 ranking algorithm. Assume that a paper [15] is selected to compute the C3 rank.

First the system extracts the references from the paper and saves them into database. At the same time, it extracts the important keywords from the paper according to occurrence of the keywords. Table 1 gives a list of these keywords with their frequency.

**Table 1. Keywords of Research Paper**

Keywords	Frequency
Science	9
Citation	7
Rank	5
Theory	4
Pagerank	3
Algorithm	3
Index	2
Application	2
Technology	2
Bibliometrics	2
Article	1

**Summary:** Web Mining is defined as the application of data mining techniques on the World Wide Web to find hidden information. This hidden information i.e. knowledge could be contained in content of web pages or in link structure of WWW or in web server logs. Based upon the type of knowledge, web mining is usually divided in three categories: web content mining, web structure mining and web usage mining. An application of web mining can be seen in the case of search engines. Most of the search engines are ranking their search results in response to users' queries to make their search navigation easier. In this paper, a survey of page ranking algorithms and comparison of some important algorithms in context of performance has been carried out. The most important component of the search engine (see Fig. 2) is a crawler also called a robot or spider that traverses the hypertext structure in the web and downloads the web pages. But before representing the pages to the user, some ranking mechanism (web mining) either in back end or in front end is used by most of the search engines to make the user search navigation easier between the search results. In this paper, a survey of various page ranking algorithms has been done and a comparison is carried out. This paper is structured as follows: in section 2, web mining concepts, categories and technologies have been discussed. Section 3 provides a detailed overview of some page ranking algorithms and section 4 discusses the limitations and strengths of each algorithm discussed. The size of WWW is growing rapidly and at the same time, the number of queries, the search engines can handle has grown incredibly too. With increasing number of users on the web, the number of queries submitted to the search engines are also increasing exponentially. Therefore, the search engine must be able to process these queries efficiently. Thus, some web mining technique must be employed in order to extract only relevant documents from the database and provide intended information to the users. To present the documents in an ordered manner, *Page ranking* methods are applied, which can arrange the documents in order of their relevance, importance and content score and use web mining techniques to order them. Some algorithms rely only on the link structure of the documents i.e. their popularity scores (web structure mining), whereas others look for the content in the documents (web content mining), while some use a combination of both i.e. they use links as well as content of the document to assign a rank value to the concerned document. Some of the common page ranking algorithms have been discussed as follows.

**Fig 4: Summary of the Research paper**

**Table 2 Similarity Score of Research Paper**

Paper Title	Similarity Score
Focused Page Rank in Scientific Papers	0.8
Google Scholar's Ranking Algorithm: An Introductory Overview	0.53
Prediction and Ranking Algorithms for Event Based Network Data	0.03
Ranking Authors in Digital Libraries	0.38
On the Pagerank Algorithm for the	0.61
Sum of Similarity Score	2.35

On the basis of these keywords, summary of the paper is generated. The lines of the summary could vary according to user's requirement. By default, the summary of the paper is of 20 lines. Fig 4 shows the summary of this paper.

Now, to calculate the C3 rank of the selected paper, system extracts the research papers having citations to it.

Following research papers cited the selected paper:-

- Focused Page Rank in Scientific Papers Ranking
- Google Scholar's Ranking Algorithm: An Introductory Overview
- Prediction and Ranking Algorithms for Event Based Network Data
- Ranking Authors in Digital Libraries
- On the Pagerank Algorithm for the Articles Ranking

The similarity scores obtained after comparison of summaries of these papers with summary of selected paper are shown in the Table 2:

Then, to calculate the C3 rank of uploaded paper, find mean of total similarity score:-

$$\text{C3 Rank} = \frac{\text{Total Similarity Score}}{\text{Total number of cited paper}}$$

$$= \frac{2.35}{5}$$

$$= 0.47$$

At last, the final rank of the paper is 0.47. This rank is calculated by measure the similarity between the selected paper and papers which cited it.

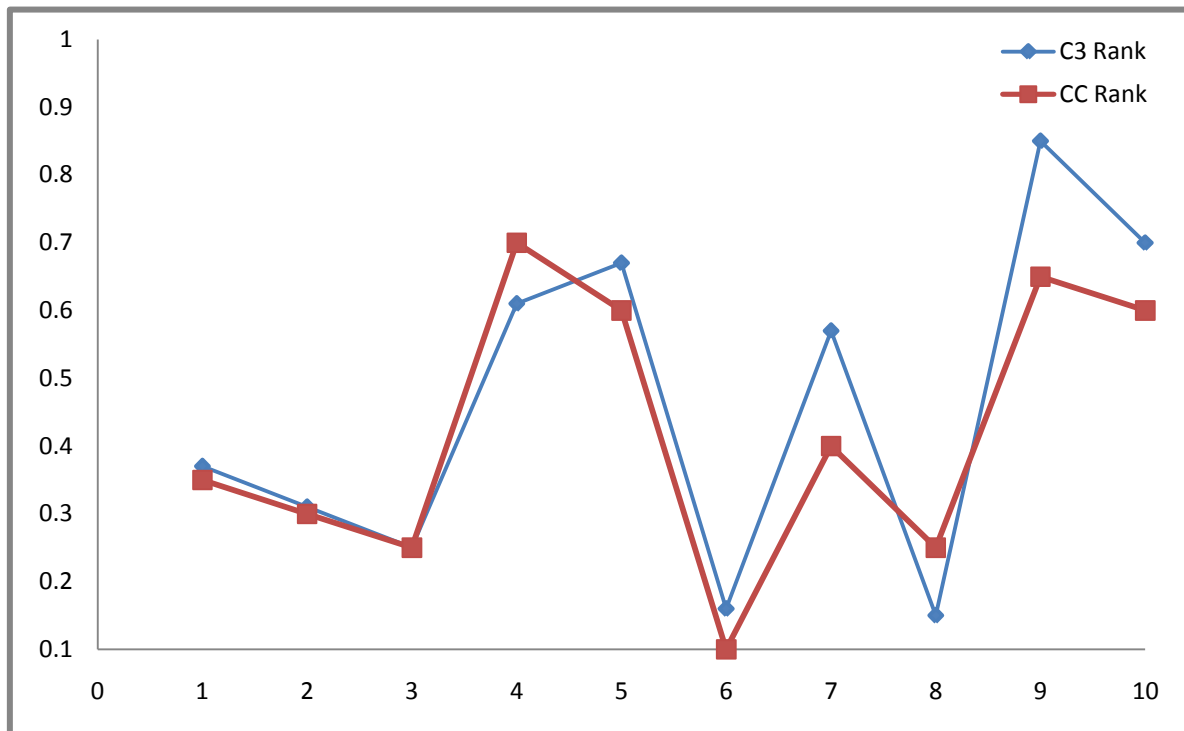
## 5. RESULT ANALYSIS

In this section, the results of the proposed *C3 ranking algorithm* are compared with the existing *Citation Count (CC) ranking algorithm* with the help of graph. We take ten papers for which comparison is done. The CC rank and C3 rank of the ten papers as return by the implemented system are shown in the table 3. CC rank ranges from 1 to 20 but C3 rank ranges from 0 to 1. So, normalization of CC rank is done by dividing the CC rank by 20 so that CC rank lies between 0 to 1.

It may be noted that, the rank of CC and C3 algorithm is very different. Analysis of these two algorithms is done by plotting a graph between their variations. The graph is shown in Fig. 5.

**Table 3 Result of two Ranking Algorithms**

PaperID	CC rank	Normalized CC rank	C3 rank
1	7	0.35	0.37
2	6	0.30	0.31
3	5	0.25	0.25
4	14	0.70	0.61
5	12	0.60	0.67
6	2	0.10	0.16
7	8	0.40	0.57
8	5	0.25	0.15
9	13	0.65	0.85
10	12	0.60	0.70



**Fig. 5 Graph Representation of above Results**

From the graph, it can be observed that C3 rank is very affected by the relevancy of the paper. If a paper A is cited by another paper B and paper B is not related to the field of paper A, the CC rank of paper A is increased by 1 due to count of citation but the C3 rank is not increased by 1. C3 rank checks the relevancy of both paper A and B and increase the rank of A by some relevancy factor if B is related to field of A.

## 6. COMPARISON WITH OTHER RANKING ALGORITHMS

In this section, the comparison of proposed C3 ranking with two existing ranking methods i.e. Citation count ranking and Content based ranking is given. The comparison is done on the basis of different criteria i.e. basic description, type of mining, processing time, space requirement, degree of

relevance, input parameters and scanning options. Table 4 shows the comparison between these three algorithms.

From this comparison table, it can be observed that the proposed algorithm (**C3 Ranking**) is better than other two algorithms. In C3 ranking algorithm, ranking is done on the basis of links as well as content of the publication. So, this algorithm performs both type of mining techniques simultaneously i.e. structure and content mining. Also content based ranking algorithm gives a poor performance as it needs to scan the whole paper for ranking but C3 ranking algorithm improves the performance by using already created summary of the paper for ranking

**Table 4: Comparison of three different algorithms**

Criteria for comparison	Citation Count Ranking algorithm	Content Based Ranking algorithm	C3 Ranking Algorithm
<b>Basic Description</b>	Rely only on links	Rely only on content of the paper.	Rely on links as well as content of the paper.
<b>Type of Mining</b>	Structure Mining	Content Mining	Combination of Structure and Content Mining
<b>Degree of Relevance with Query</b>	Does not check the relevancy	Checks the relevancy	Checks the relevancy
<b>Different Scanning Options</b>	No Scanning	Scans the full Paper	Scans only the Summary of the Paper
<b>Processing Time</b>	Low	High	Medium
<b>Space Requirement</b>	Low	High	Medium
<b>I/P Parameters</b>	Backlinks	Paper	Backlinks and Summary

## 7. CONCLUSION

In this paper, a new ranking mechanism has been proposed. Many researchers have proposed their different ranking algorithms. In those algorithms, ranking is done either on the basis of content of the paper or citations to the paper. So, these algorithms do not provide relevant results. But in the proposed mechanism, rank of the paper is calculated on the basis of citations to the paper and content of the paper. The ranking algorithm checks the relevancy between the publication and the papers which cited the publication. Instead of scanning the whole paper, summary of the paper is used to compute the relevancy score. Due to this, it saves space and time efficiently.

## 8. REFERENCES

- [1] Naresh Barsagade; *Web Usage Mining And Pattern Discovery: A Survey Paper*. CSE 8331,2003.
- [2] M. Krishnamurthy; *Open access, open source and digital libraries: A current trend in university libraries around the world*. A General Review, Emerald Group Publishing Limited.
- [3] Sumita Gupta, Neelam Duhan, Poonam Bansal; *A comparative study of page ranking algorithms for online digital library*. International Journal of Scientific & Engineering Research, Volume 4, Issue 4, April 2013.
- [4] Neelam Duhan, A. K. Sharma, Komal Kumar Bhatia; *Page Ranking Algorithms: A Survey*. 2009 IEEE International Advance Computing Conference (IACC 2009) Patiala, India, 6–7March 2009.
- [5] Jaroslav Pokorný, Jozef Smizanský; *Page Content Rank: An Approach to the Web Content Mining*.
- [6] Joeran Beel, Bela Gipp; *Google Scholar's Ranking Algorithm: The Impact of Citation Counts (An Empirical Study)*. In Rcis 2009: Proceedings of The IEEE International Conference on Research Challenges In Information Science, 2009.
- [7] L. Marian, M. Rajman; *Ranking Scientific Publications Based on Their Citation Graph*. Master Thesis, CERNTESIS, 2009.
- [8] L. Marian, J. Yves LeMeur, M. Rajman, M. Vesely; *Citation Graph Based Ranking in Invenio*. ECDL, pp. 236-247, 2010.
- [9] Lawrence Page, Sergey Brin, Rajeev Motwani and Terry Winograd; *The pagerank citation ranking bringing order to the web*. technical report, computer science department, Stanford university; 1998.
- [10] Sergey Brin and Larry Page; *The anatomy of a Large scale Hypertextual Web Search Engine*". In Proceedings of the Seventh International World Wide Web Conference, 1998.
- [11] Kleinberg J.; *Authorative Sources in a Hyperlinked Environment*. Proceedings of the 23rd annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1998.
- [12] Palakorn Achananuparp, Xiaohua Hu, and Shen Xiajiong; *The Evaluation of Sentence Similarity Measures*.
- [13] Landauer, T.K., Laham, D., Rehder, B., and Schreiner, M.E. (1997); *How Well Can Passage Meaning Be Derived without Using Word Order? A Comparison of Latent Semantic Analysis and Humans*. in Proc. 19th Ann. Meeting of the Cognitive ScienceSoc., 412-417.
- [14] Achananuparp, P., Hu, X., Zhou, X., and Zhang, X. (2008); *Utilizing Sentence Similarity and Question Type Similarity to Response to Similar Questions in Knowledge-Sharing Community*. To appear in Proceedings of QAWeb 2008 Workshop, Beijing, China.
- [15] Hacene Ait Haddadene, Hakim Harik And Said Salhi; *On the Pagerank Algorithm for the Articles Ranking*. Proceedings of the World Congress on Engineering 2012 Vol I WCE 2012, July 4 - 6, 2012, London, U.K