A Novel Algorithm for User Feedback Based Page Ranking for Research Field

Sandhya Verma*1 and Amit Kumar Manjhvar*2

* Research Scholar, CSE/I.T., Madhav Institute of Technology and Science, Gwalior, India
# Assistance Professor, CSE/I.T., Madhav Institute of Technology and Science, Gwalior, India

Abstract—Page ranking is a technique to find the ranking of the web pages on the basis of importance of web pages. Page ranking in research field is a new technique proposed to derive the value and importance of the research paper and the particular type of work done in the paper regarding the research. The search engine utilizes the page rank algorithm according to dynamic query and information retrieval based on authors name and their title of the work. The implementation of the proposed algorithm uses various new factors like dynamic retrieval, citation and number of downloads based work and user feedback based updation of the rank of the paper for the author. The results show that the proposed technique is a new one and works for all scenarios of the research and development field.

Index Terms—Web mining, usage mining, recommendation System, page ranking, citations.

I. INTRODUCTION

A large portion of the deep web is database based, i.e., for many search engines, data encoded in the returned result pages come from the underlying structured databases. Such type of search engines is often referred as Web databases (WDB). A typical result page returned from a WDB has multiple search result records (SRRs). Each SRR contains multiple data units each of which describes one aspect of a real-world entity.

With the explosive development of internet, the amount of information has become more and more, and beyond scope of people’s acceptance. Recently, many individual recommendation systems and personalized services provide suitable information to different people. These systems and services are based on the user model. And to build a proper user model, the key step is to extract user interests accurately. So nowadays how to get user interests from the history of user behavior is an important issue.

User behaviors are important factors which can give a great contribution degree to user interests. Some researchers have investigated in extracting user interests by considering user behaviors[1] proposes that users will spend a great length of time reading those articles they are interested in [2] investigates the behaviors such as printing, saving, scrolling and bookmarking, they consider that if user has these behaviors when browsing a document, they must have some Interests in it. But most of them don’t consider enough behaviors such as relative browsing time, printing.

This paper proposes user interaction based page rank updation for authors in research field. The research field has been used as a very few focus has been put in this field. The different kinds of contribution degrees from user behaviors to user interests are also analyzed in this paper.

This paper has the following contributions:

- Dynamic retrieval of author with paper name
- Citation based updation
- Overall weighted page rank
- Time utilization
- User based page rank updation
- Greater the number of citations, the higher the author is known and experienced.
- Also the page rank of the related links lead to the result that all the related pages and the weighted page ranking concept can help a person know about the page rank of the paper.

II. RELATED WORK

There has been considerable work in the field of academic research. Most of the work is on researcher profiling, which aim at ranking the authors. The work by Quinkun Zhao et al. [3] is one such work, which studies the relationship between authors using community mining techniques. Other work is Arnet Miner [4] which ranks the authors on h-index [5] and conferences on impact factor [6].

Our main focus here is to rank the research papers. The rankings for conferences and authors are derived from the research paper ranks. Most work on academic research uses number of citations as the metric. However, it is quite intuitive that this metric ignores the importance of the quality of citations, taking into consideration only the quantity of citations. The metrics like h-index, g-index and impact factor are based on the number of citations, and hence would not give correct results in all cases. We use a modified version of the PageRank algorithm [7], which considers the quality of the citing paper to rank the paper being cited. Also, unlike in h-index, here we need not remove the self-citations because our main focus is ranking research papers. Author ranking is taken care of by the research paper scores. The metric impact factor has another drawback that it uses only the citations from only the last two years, which simply neglects the effect...
of papers on comparatively older papers that they cite. It is to be noted that the metrics based on number of citations are highly discipline-dependent. This effect is diminished by the fact that if a field has more number of citations in general than the other fields, the division by number of citations during contribution of a citing node works in the same way as normalization works.

There has been some work on academic research using PageRank algorithm like the work by Ying Ding et. al [8], where author co-citation network is used to rank the authors.

III. PROPOSED WORK

The score calculation of the research papers help to know about the page ranking of the author’s web page. The quality, popularity of author, bloom in particular field/topic, citations, new theory giver authors are more influential factors to find the rank of the authors. So for the author scores, we use both the original page ranking and the related links page ranking for all the links in which it is shown by the browser. Some straightforward uses can be:

- Comparing two or more authors of same research field.
- Comparing the access time to access the web pages of the authors.
- Comparing the page ranking of the pages of the authors to find their popularity.

PROPOSED MODEL

PROPOSED ALGORITHM PSEUDO-CODE

The proposed algorithm works on dynamic database of the authors and gives the dynamically found links of the author with their page ranks. The final page ranks are calculated based on few parameters which affect the page rank. The pseudo-code for the proposed algorithm’s working can be described as below:

Input: Author name, paper title
Output: Page Rank, Citations, access time
Algorithm:
1. Input the author name + “Google Scholar Citations” + paper_title
2. Set time = 0, link = 0, page_rank = 0
3. Write the output details in the file using bufferedWriter by taking data using the function getDataFromGoogle() and crawler().
4. Finding all the links and writing them in a file named links.txt along with each page’s page rank, citations, outlinks and access time.

FIG 3.1 PROPOSED FRAMEWORK
5. Top 10 links having high page ranks regarding the author are selected and page rank are calculated based on the following formulas:

a. \[ PR_{net} = \frac{PR_1 + PR_2 + \ldots + PR_n}{n} \]

b. GetCitations()
   
   \[ NCit = cit_1 + cit_2 + \ldots + cit_n \]

c. AccessTime

Access time is calculated for each page retrieved for the author

\[ \text{Set } A_{\text{Max}} = 20.0 \]
\[ \text{If(access_time} > A_{\text{Max}}) \]
\[ \text{Access_time} = A_{\text{Max}} \]
\[ \text{Else} \]
\[ \text{Show access_time for each page.} \]

12. Now user based page rank updation is performed, in which the rating and feedback from the user is taken in form of ratings.

13. Page rank is updated based of the scores provided by the user.

14. End.

IV. RESULT ANALYSIS

The results are generated based on the authors and their citations. The results are compared for both the base and proposed codes.

The improvements done in the proposed work are:

1. Dynamic page ranking algorithm.
2. Updation in page ranking of author based on number of citations.

A comparative study is shown between the base and proposed code page ranks that are evaluated based on the proposed algorithm. Also a screenshot of the derived web links of the user given author is shown so as to show that it is dynamic.

Step 1: Input the author name and find the links related to the author. The coding is done in java, thus the results are shown in command prompt. The screen shows as below:
The results show the updated page rank in the proposed work. The graph below shows the comparative study.

1. Comparative results of Page Rank Updation

![Graph showing comparative results of Page Rank Updation](image)

2. Comparison of the page ranks given by the users to various authors.

![Graph showing comparison of different ratings by users](image)

**REFERENCEs**


