Int.J.HIT.TRANSC:ECCN.Vol.3: Issue 2A(1) (2017) Page 21-33 ISSN: 0973-6875

Available Online at www.hithaldia.in/locate/ECCN

ORIGINAL CONTRIBUTION

Web Page Ranking Based On User Query

Pranab Goswami*, Moumita Mantri and Manasija Bhattacharya

Depertment of Information Technology, Haldia Institute of Technology, Haldia, India

(Received Date: 20th August, 2017; Acceptance Date: 30th September, 2017)

ABSTRACT

Web page ranking is a challenging task in the field of information retrieval research. Different page rank algorithm likes hits algorithm, link based algorithm are commonly used to rank the web pages. To produce more relevant and important results we have implemented scope to test and check relevancy of a page and at the same time its importance .The relevancy of a page is judged on basis of the user query. We match and do all the operations required to know the query of the user what it intends to and what does it want and then we crawl the webpage's by having an index to each webpage and then match its keyword or the frequency of the occurrence of the highlighted portion of the user's query with that of the keywords of a webpage. This paper mainly focuses on the relevancy and importance of webpage's along with the computational speed for the operation. This paper also focuses on giving fewer results to the user unlike other search engines those give millions of results out of which an average user only crawls about 10-20 pages average approx. This electronic document is a "live" template and already defines the components of your paper.

Key words: Page; Page Ranking; User Query; Damping Factor; Web Page

1. INTRODUCTION

The aim of this project paper is to develop a model for a web page ranking to produce relevant, important results in quick time. This project also aims in rendering only few outputs which are useful for the user as per its query. This paper also implements the existing algorithm such as Google's page rank algorithm, hits algorithm, link based algorithm. But we are trying to do in this project is we are introducing customization in our search engine. This makes search engine more efficient to produce results. We will classify the web pages on basis on their keywords obtained into various subjects and then group all web pages as per as their category. The other objective of the project paper is to shift the focus from result oriented to relevancy and importance of the web page. Users query the web for different purpose and hence to know exactly what the user wants is the most difficult task. In recent search engine the focus are not only given to relevancy bur also to authoritativeness. That means the page must be trusted and must have a strong presence in the web. The algorithm we have implemented in this project checks relevancy of page, crawls each web page, index the page, implements link based algorithm to give link based ranks and then if new page is found it checks whether found page is relevant or not and then finally all the web pages are stored in data structure and then ranked again as per the no of visits and then the final result is rendered back to the user.

2. PREVIOUS WORK

A. The Page Rank Algorithm

In this approach we rank pages based on the number of pages that links to it .Consider a web page p(A) and p(T1.....Tn) page has got link to page p(A) then the total rank of the page p(A) will be the rank of total summation of the pages of p(T1.....Tn). In this approach we introduce damping factor denoted by (d) such that it gives the whole result in form of probability. The sum of all results will always be 1. The damping factor is introduced so that a time will come when user will not click any further links as he will get bored. The damping factor lies between 0

*Corresponding Author: pranab.goswami07@gmail.com

to 1 and in current scenario it is assumed to be (d=0.85). PR (A) = (1-d) + d (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn)). The rank of page p(A) has been calculated by the total number of inbound links pointing to page and recursively calculating the pointing page that is p(T1.....Tn) pages rank. C(T) is the total number of out bounds from the page p(A). In this approach the page rank of a page is not only influenced by the rank of inbound pages but also by the number of outbound pages from that page. This means that the more outbound links the page p(T) has the lesser rank page p(A) will have. Below is the justification of damping factor and why it has been chosen and how this mathematical formula was developed

B. The Random Surfer Model

The Page Rank as a model of user behavior, where a surfer clicks on links at random with no regard towards content. The random surfer visits a web page with a certain probability which derives from the page's Page Rank. The probability that the random surfer clicks on one link is solely given by the number of links on that page. This is why one page's Page Rank is not completely passed on to a page it links to, but is divided by the number of links on the page. So, the probability for the random surfer reaching one page is the sum of probabilities for the random surfer following links to this page. Now, this probability is reduced by the damping factor d. The justification within the Random Surfer Model, therefore, is that the surfer does not click on an infinite number of links, but gets bored sometimes and jumps to another page at random. The probability for the random surfer not stopping to click on links is given by the damping factor d, which is, depending on the degree of probability therefore, set between 0 and 1. The higher d is, the more likely will the random surfer keep clicking links. Since the surfer jumps to another page at random after he stopped clicking links, the probability therefore is implemented as a constant (1-d) into the algorithm. Regardless of inbound links, the probability for the random surfer jumping to a page is always (1-d), so a page has always a minimum Page Rank. The Page Ranks then form a probability distribution over web pages, so the sum of all pages' Page Ranks will be one

C. Proposed Page Rank Algorithm

Let us consider four pages p(A), p(B), p(C) and p(D). We calculate the rank of page PR of p(A) where damping factor d=0.85 [8].

PR(A)=1-d+d*(PR(D)/2).

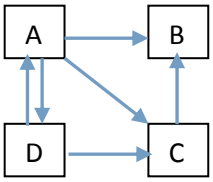


Figure 2.1: Page Rank Algorithm

The logic behind this equation is the number of outbound links from page p(D) is 2 that is to page p(C) and p(A) .Hence, we have divided it by 2. The damping factor is multiplied with page rank of all the inbound pages .The inbound page in this case is p(D).

PR(A)=1-d+d*(PR(D)/2).

Similarly, the rank of the page p(D) is PR(D)=1-d+d*(PR(A)/3). The reason is same as stated above. The number of outbound link from page p(A) is three those are to page p(D),p(C) and p(B).

The page rank of page p(B) is calculated similarly.

PR(B)=1-d+d*(PR(A)/3+PR(C)).

Page p(C) has got only one outbound link that is only to page p(B).

Now we calculate the last page p(C) .

PR(C)=1-d+d*(PR(A)/3 + PR(D)/2)

The number of outbound link from page p(D) is 2 that is to page p(A) and p(C). Four equations and four results which are shown below.

Equation 1: PR(A)=0.15+0.85*(PR(D)/2)

Equation 2: PR(D)=0.15+0.85*(PR(A)/3)

Equation3: PR(B)=0.15+0.85*(PR(A)/3 + PR(C))

Equation 4: PR(C)=0.15+0.85*(PR(A)/3 + PR(D)/2)

Now solving the above equations the results are appended below:-

Result 1: PR(A)=0.243

Result 2: PR(B)=0.483

Result 3: PR(C)=0.311

Result 4: PR(D)=0.218

We can see the results and can be found that page p(B) has the highest rank. The order of the web pages based on the calculation is as follow:

p(B)>p(C)>p(A)>p(D)

We iterate the above equations over a twenty times by implement a simple C program. The iterative result is illustrated in Table1. Each page is assigned a starting Page Rank value of either 1 or 0. We have considered the starting value is 0 for our experiment. The output says that the result is unchanged after little iteration.

C. Different Parameters to Rank Page

IR SCORE (Information Retrieval): The query submitted by the user is often checked by the search engine in various ways. The query is matched against the meta tag of the web page , the description tag of the web page and some time the body content are also checked .Images , graphics ,anchor text ,link containing text are few different entities to check .This called an IR of an page which is called Information Retrieval. An IR score is mainly calculated by the anchor text of inbound links of a page, which is weighted by position and accentuation of the search term within the document. This way the relevance of a document for a query is determined. The IR-score is then combined with PageRank as an indicator for the general importance of the page. To combine the IR score with PageRank, the two values are multiplied. It is obvious that they cannot be added, since otherwise pages with a very high PageRank would rank high in search results even if the page is not related to the search query. The IR score plays a very important role to rank the WebPages.

Page Factors: There are different On-Page and Off-Page factors to rank a page

The On-Page Factor:

1) Proper keyword in the title or meta tag

2) Keyword in the URL (uniform resource locator)

3) Keyword in the domain

4) Quality of the html code

5) Page refreshes

The On-Page factors basically focuses on the keywords, title, body content and quality of the code used. It helps to minimize the work of search engine to detect relevancy of the page. In this way the search engine collects few web pages on the page factors and then implements the page rank algorithm. The IR score comes next into the action where it is multiplied with the rank of the each page. It ensure that the webpage have a strong presence over the web and have authoritativeness. It also ensures that the domain is registered and not expired. This process is carried at the initial level. The repetition of the keywords is removed and proper algorithm is designed to find the frequent of the keywords.

The Off Page Factor includes:

1) The high rank page

2) The anchor text of the inbound links

3) Links from authority sites

4) Site age (stability)

The off page factors focuses on the illegal things done to optimize their web page. The high page rank must not get an inbound link from a very low rank page where the rank difference is too high. Such web pages are treated as Spam web pages. Spam web pages must be removed or should not be rendered back to the user as result. Link buying is another important factor where a page tends to increase its number of link by creating different pages and giving link to its site. Links from bad neighbor is not encouraged and treated as spam.

D. The Effect of the Inbound Link

Now consider a case where an additional page which has got many inbound link and has got only one outbound link. What we will think in such case, confused. Say page p(X) which has got 10 inbound links and its rank is PR(X)=10 points a page p(A) which has got rank of PR(A) say 5. Now in such case what could we conclude? Now there are two case studies that have been done from our side.

Let us analyze the effect of inbound links on the rank of page. Considering the figure 2 where we have four pages we create a page p(X) which links to page p(A). Than we see how the output varies and why we need a different approach than the traditional Google's approach.

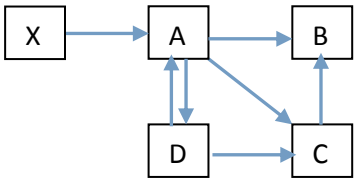


Figure 2.2: Inbound Link

Now we have calculated the values of each page using the formula shown below.

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

Let us consider the new page p(X) has its rank PR(X)=10 (say 10 as it is easier for calculation) has it has got no inbound links but we assume it actually is.

Equation 1: PR(A)=0.15+0.85*(PR(D)/2+PR(X))=0.15+0.85 *(PR(D)/2+10)

Equation 2: PR(D)=0.15+0.85*(PR(A)/3)

Equation 3: PR(B)=0.15+0.85*(PR(A)/3 + PR(C))

Equation 4: PR(C)=0.15+0.85*(PR(A)/3 + PR(D)/2)

Now solving these four equations we get the following output PR(A)=9.906, PR(B)=6.538, PR(C)=4.213, PR(D)=2.956 (cumulative rank = 23.613). We compare these result with previous found results: PR(A)=0.243, PR(B)=0.483 PR(C)=0.311, PR(D)=0.218 (cumulative rank = 1.255).

Now we will try to put justification for the above results. Initially PR (A) was 0.243; the new page p(X) has done some additional advantage to increase its rank. Now due to p(X) the rank must be increased by a factor d*PR(X)/C(X) where C(X) is the total outbound link of page p(X). So here C(X)=1 as it has got only one outbound link . Now calculating the value 0.85*10=8.5 hence the PR(A) must be at least be 0.243 + 8.5 = 8.743 but the value we got is Now where does this extra (9.906.-8.743) 1.163 came . This is due to fact that not only page p (A) has been effected but all other pages are too. Now the page again points back to page p (A) as we can see page p(D) points to page p(A). Hence the result have bounced back to page p(A) and increased its rank.

The page rank of p (A) has got approx the page rank of p(X). Now we are trying to see it with more practical orientation when such things are possible. Now we will do some case study to analyze the fact in more details where random surfer model will be considered.

Case Study 1: Consider user write a query such as "How to program in java using MySql connection "Now it may be so that page p(X) has some related keywords and we can say it is an relevant page based on the user's query. Now page p(X) points to page p(A) . Now there could be many possible ways why page p(X) has a link to page p(A).Listed below are some possibilities.

Possibility 1: The page p(X) might have link for advertisement, some other videos, files, and websites with some different contents. But the reason is page p(X) is mainly a programming website. So now if we rank page p(A) which comes around 9.906 and if there are more inbound links then page p(A) must be rendered before the page p(X) back to the user as query result . Hence this is the limitation of this particular approach.

Possibility 2: Now the second possibility could be that the page p(A) have some relevant information based on the user's query . But the

question is how much relevant, how far relevant whether more than page p(X) or not? Now answering this question could be again based on the probability of the random surfer that why did he clicked this link which points the page p (A). There could be more than one reason. Might be the user was not satisfied with content of information of page p(X) and hence clicked the link page p (A) in order to get some more relevant information. Now when reaches the page p(A) it depends again whether he satisfied or not .Now what if the user still does not satisfied with the content. What if he founds some irrelevant information? But there could be also the possibility that the user may get satisfied with the content of the new page p (A).

Case Study 2: Now after writing the query user gets the page p(X) and then the following link it has. He clicks the link p (A) and then finds link for page p(C), p (D), p (B). Now he will move to visit the page B, C, D if he is not satisfied with the answer or else just randomly .Again these things will iterate still he stops clicking link. But when you are developing your search engine where should you be more focused, on the results or the user satisfaction. Of course user satisfaction in search engine merely a chance of probability because you cannot know what can satisfy a user request. Considering the above case study we will propose our novel theory for more practical approach so that it can satisfy the need of the user.

E. The R-factor

We will introduce a new factor called R factor which means relevancy factor. The R factor will be in scale from 0 to 1 reason being the same to make the sum 1.The R factor will be determined by the keywords, content of the page against the query of the user. Determining the relevancy is very difficult as we have to fix some parameters to give the relevancy factor to each page. We have given this proposal keeping in mind not only the content based results are important but the giving relevant page information are also important .We want that the user must click the minimum no links to get his request. So our proposal is that minimum effort put by a user to get the relevant information.

Calculation of the R factor : Now consider a situation where user writer's a query as "How make a project in net beans with tomcat server". Now by just seeing the query it would be really tough to identify which words must be given more importance. Now the query is matched against the keywords, anchor text, body contents, images, and graphics of each web page. Now the web pages which have similar content are rendered. Suppose WebPages such p(Y), p(Z) and p(X) has got the same content . Now again considering the figure 1.2 where p(X) points to p (A). Now before doing the calculation our first task would be find wither the new page p (A) has got the relevant information based on user's query or not this is again checked by matching the keywords as discussed earlier.

Now the impertinent question is how shall we give the value of the R factor to each page ?Now suppose the web page consist of the words such "tomcat server deployment "and no keywords mapped against project .Now after a brief analysis words stress must be given to "project, net beans, tomcat server "Now out of those only two words are mapped hence R factor of page p (A) is = 2/3 =0.666.

F. The Effect of the R-factor in Page Rank

Now suppose user writer's the same query such as "How make a project in net beans with tomcat server". Now considering the first case and the diagram we will analyze the result and its effect. After writing these query let the corresponding page found is p(D) (you can take any page for consideration).

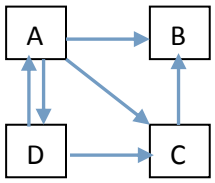


Figure 2. 3: In and out bound Link

PR(A)=0.15+0.85*R(A)*(PR(D)/2), where R(A) is the relevancy factor of page p(A). ……..Equ(1)

PR(D)=0.15+0.85*(PR(A)/3) in page p(D) R(D) is 1 as it was the first page rendered. ……. Equ(2):

PR(B)=0.15+0.85*R(B)*(PR(A)/3 + PR(C)) Equ (3)

PR(C)=0.15+0.85*R(C)*(PR(A)/3 + PR(D)/2). ……Equ(4)

Now for assumption we have considered the R (A)=0.66 R(B)=0.54 and R(C)=0.85. Now solving the four equations we will get the following results. Result after implementing the R-factor. PR(A)=0.208, PR(B)=0.308, PR(C)=0.275, PR(D)=0.209 (Cumulative sum= 1)

Now comparing it with the above result without the R-factor is

PR(A)=0.243, PR(B)=0.483, PR(C)=0.311, PR(D)=0.218 (Cumulative sum = 1.255)

Compare page rank with and without implementation of R-factor.

With R-Factor: p(B)>p(C)>p(D)>p(A)

Without R-Factor: p(B)>p(C)>p(A)>p(D)

G. The Effect of R-factor and Inbound Link

Now we try to analyze the effect combined inbound link and R-factor. Please note that for different assumption ranking of pages will differ and it completely depends on the page which has got maximum number of keywords hit. The R-factor is introduced so that user does not have to surf lots of pages to get their information. We want that sure should get their desired result at the earliest and most convenient way. They do not have to click unnecessary links to visit different pages to get their result. The surfer model has been kept into consideration while implementing the so called the R-factor

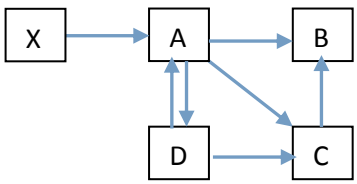


Figure. 2. 4: R factor of inbound link

Now again taking the assumption let page p(X) was found after the user submit its query. Now we will see what could be the result .We will take R(A)=0.66 R(B)=0.54 R(C)=0.85 R(D)=0.5

Now the equation will changed to the following.
Equation 1: PR (A)=0.15+0.85*R(A)*(PR(D)/2+PR(X)).= 0.15+0.85*0.66*(PR(D)/2+10).

Equation 2: PR (D)=0.15+0.85*R(D)*(PR(A)/3) = PR(D) = 0.15+0.85*0.5*(PR(A)/3).

Equation 3: PR (B)=0.15+0.85*R(B)*(PR(A)/3 + PR(C)). = 0.15+0.85*0.45*(PR(A)/3 + PR(C)).

Equation 4: PR(C) = 0.15+0.85*R(C)*(PR(A)/3 + PR(D)/2).=0.15+0.85*0.85*(PR(A)/3 + PR(D)/2).

Now solving the four equations we will get the following output. PR(A)= 6.042, PR(B)= 1.978, PR(C)= 1.968, PR(D)= 1.005, cumulative sum = 10.993. Now compare the result of inbound effect without the R-factor. PR (A)= 9.906, PR(B)= 6.538, PR(C)= 4.213, PR(D)= 2.956, cumulative sum = 23.613

Now we can clearly analyze the effect of the R-factor in the webpage ranking. It varies from page to page, keywords matched and then the link it has. We can conclude that this new approach to the introduction of the R-factor can change the scenario of search engine in future. Clearly it is based on the ease of human being to get their result. Practically speaking who want to wait for long time to get result surf pages across the web. We live in a world where time is the most precious thing where every human wants things as fast, accurate and relevant as possible. Now keeping all these things in mind we will generate only few results but they will be the most relevant and important information required by the user. We have made many assumptions that clearly provide us the fact that R-factor will vary from the query submitted by the user we will try to focus more on what user wants than what he gets. To know the intention of human being is the biggest challenge in this 21st century. We just don't want to build a search engine that's

sole role to provide information but in fact we want to build a search engine that can understand the human intention.

H. The Effect of Outbound Link

As we have seen the inbound links plays a very crucial and very important role in the ranking of web pages. The same way outbound links plays a very important role in page ranking. We take two different systems and then we prove that we cannot derive a formula that can satisfy all the test system. Hence a generic approach cannot be made as webpage system always varies. In the first system we will consider two website named as WEB X and WEB Y.

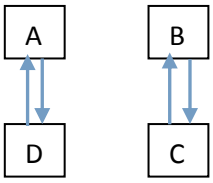


Figure 2.5: a(Web X) Figure 1.5a(Web Y)

The WEB X has two web pages named P(A) and P(D) which have got only eternal link where as the WEB Y has got two web pages named page P(B) and page P(C). Now we try to create a link from page P(A) of one system to page P(B) of the other system.

Now we create equation for above situation for each system for system WEB X.

Equation 1: PR(A)=0.15+0.85*(PR(D))

Equation 2: PR(D) = 0.15+0.85*(PR(A)/2);

For System WEB Y

Equation 3: PR(B) = 0.15+0.85(PR(A)/2+PR(C)).

Equation 4: PR(C) = 0.15+0.85(PR(B)).

Now we try to solve the four equations to get the result and the output will be as follows: PR(A) = 0.434 , PR(D)= 0.334 , PR(C) = 1.563 PR(B)=1.66. Now by the cumulative sum of the rank of system WEB X=0.434+0.334=0.768, and the cumulative sum of the rank of system of WEB Y= 1.563+1.66=3.223. The cumulative rank of both system=0.768+3.223=3.991. Now we take another system, the first is a

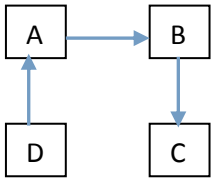


Figure 2.6: Combined of X and Y

closed system and then we will prove that close system practically does not means anything.

Now we will take another system which is an open system where the internal links within a website will no be taken into consideration. We can note that page p(D) has got no inbound links.

Now we write equation for each system again for WEB X and WEB Y

For system WEB X

Equation 1: PR(A)=0.15+0.85*PR(D)

Equation 2: PR(D)=0.15+0 (As it has got no inbound links)

For system WEB Y

Equation 3: PR(B)=0.15+0.85*PR(A)

Equation 4: PR(C)=0.15+0.85*PR(B)

Now after solving the four equations we get the following results.

PR(A)=0.277 , PR(D)=0.150 , PR(C)= 0.477 , PR(B)=0.385

Now by the cumulative sum of rank of the system WEB X= 0.277+0.15=0.427

And the cumulative sum of rank of the system WEB Y=0.477+0.385=0.863

The cumulative rank of both system=0.427+0.862=1.289

Hence there is a difference in the cumulative and individual rank of system when both the open and the closed systems were taken into consideration. We may think that system can be closed but in more practical aspect they are not. So from now onwards we will deal only with the open system and then implement different approach to find the ranks of pages. The reason why we are not considering closed because the website has many pages which are in hierarchical form. so the is no

1.10.477, PR(A)= 0.385. Hence we can justify that if system is open and initial keywords matched from one website than all the pages remain unaffected by the R-Factor.

H. The Effect of Number of Pages

Now let us again consider a situation where a website has a number of pages and when an additional page is added up what could be the outcome. When we talk about a website then it's a closed system where each pages refers each other for some reason. Again we will take two different website and then we will add page in one of those website to see the result.

Now let us take a website and create the equations required.

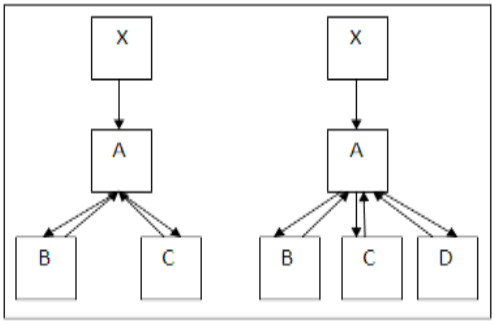


Figure 2.7: Add p (d))

The website X had initially pages p(A),p(B),p(C) and then an additional page p(D) has been added up. Now formulate these equations at the initial condition.

Equation 1: PR (A) = 0.15+0.85*(PR (B) + PR(C)).

Equation 2: PR (B) = 0.15+0.85*(PR (A)/2).

Equation 3: PR (C) = 0.15+0.85*(PR (A)/2).

Now solving the equation we get the following results.

PR (A) = 1.457, PR (B) = 0.769, PR(C) = 0.769, Cumulative sum = 2.995. Now considering the second condition and formulating equations for it.

Equation 1: PR (A) = 0.15+0.85*(PR (B)+PR(C)+PR(D)).

Equation 2: PR (B) = 0.15+0.85*(PR (A)/3).

Equation 3: PR (C) = 0.15+0.85*(PR (A)/3).

Equation 4: PR (D) = 0.15+0.85*(PR (A)/3).

Now solving the above equations we get the following results.

PR (A) = 1.916, PR (B) = 0.692, PR(C) = 0.692, PR (D) = 0.692 Cumulative sum = 3.992. We can conclude that total rank of the website was increased by one. Hence now we can put a statement that when an additional WebPages are added and all the WebPages are strongly interlinked with one another than the website rank increase by one if considered it as a complete close system. Always to be close it is necessary that all the WebPages within the system must be strong interlinked.

H. The R-factor Implementation

Now we directly write down the equations by assuming some values for-Factor for each page. Implementation in the first case: Let us assume that R (A) = 0.23 and R(C) = 0.36. P (B) is the initial webpage. R (B) = 1.

Equation 1: PR (A) = 0.15+0.85*0.23*(PR (B) + PR(C)).

Equation 2: PR (B) = 0.15+0.85*(PR (A)/2).

Equation 3: PR (C) = 0.15+0.85*0.36*(PR (A)/2).

Now the results of the above equations are PR (A) = 0.235, PR (B) = 0.249, PR(C) = 0.185, Cumulative sum = 0.669. Now we will take two different websites and then add a page in one of the website. This will be implemented in open system. We have considered the figure fig 1.5 where have taken two

Different websites named WEB X and WEB Y than we have linked the page from p(A) to p(B) and then we have increased a page of web site WEB Y named as page p(E).

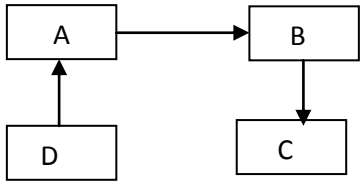


Fig 2.8: Web and Y

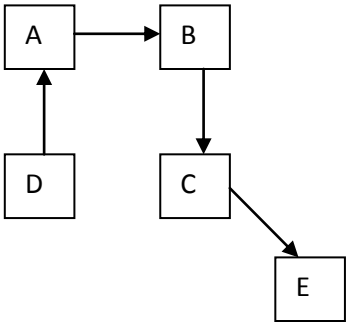


Figure 2.9 : Add E page

We had the result of the initial condition that is of fig 2.1 and that is PR(A)=0.277 PD(D)=0.150 PR(C)=0.477 PR(B)=0.385

Now we see the new generated equation computed for the fig 2.2

For system WEB X

Eq 1 PR(A)=0.15+0.85*PR(D)

Eq 2 PR(D)=0.15+0

For system WEB Y

Eq 3 PR(B)=0.15+0.85*PR(A)

Eq 4 PR(C)=0.15+0.85*PR(B)

Eq 5 PR(E)=0.15+0.85*PR(C)

Now solving the above equation we will get the following result to give result of Fig 2.2

PR(A)=0.277, PD(D)=0.150, PR(C)=0.477, PR(B)=0.385, PR(E) =0.556.

Now we can see that the page rank value remain unaffected of each website system. But the cumulative sum of the website WEB Y has increased due to the fact of the addition of a new web page. But the point must be considered that the additional page in one website has not done any change in the other website. The cumulative sum of website WEB X remained the same.

I. THE IMPLEMENTATION OF THE R-FACTOR:

Now we consider the figure 2.2 to find the paper rank by implementing the R-factor. Let us assume again that only page p(A) has got the relevant information based on the query. So the relevant factor of page p(A) is R(A) =1. The page p(D) does not takes part in the relevancy as page p(A) has no outbound link to. The rest of the pages will now have some R-Factor . Let the R-factor of page p(B) is R(B)=0.75, page p(C)= is R(C)=0.37 and the page p(E) is R(E)= 0.22. We have chosen the R-factors randomly and again we would like to mention that if the R-Factor is changed then the whole ranking of the website and its pages will change. Hence we must ensure that we have to implement proper algorithm to find the R Factor. This part has been left for the readers so that they can come out with some specific algorithm . But as we introduce this factor we are certainly looking forward to find a proper algorithm to find the R-Factor.

Now considering the above situation now we will formulate the equations.

Eq 1 PR(A)=0.15+0.85*PR(d).

Eq 2 PR(D)=0.15

Eq 3 PR(B)=0.15+0.85*0.75*(PR(A))

Eq 4 PR(C) = 0.15+0.85*0.37*(PR(B))

Eq 5 PR(E)= 0.15+0.85*0.22*(PR(C))

Now solving the above equations we will get the following result.

PR(A)=0.277 PR(D)=0.150 PR(C)=0.252 PR(B)=0.326 PR(E)=0.197

Now again we see that even after introducing R Factor and the result of the initial implementation of the R Factor does not change much.

R-factor can also be 1, in such case we can update our database to rank the page indexed with those key words. The R-Factor which we have introduced and also implemented has sole role to make search more relevant on basis of content. We want the surfer to get the needed result at the least clicks. The R-Factor is not any complicated concept, but what we have done is we have enhanced the existing algorithm of page rank algorithm to step a next level. This project also provides its own proposal for the customization of the search engine so that the

crawler has to do the minimum work. Next we will go through hits algorithm and then we will focus on how we have built our own search engine using a different approach to rank the pages.

J. THE HITS ALGORITHM

John Kleinberg a professor in the Department of Computer Science at Cornell came up with his own solution to the web search problem. He developed an algorithm that made use of the link structure of the web in order to discover and rank pages relevant for a particular topic. HITS (Hyperlink Induced Topic Search) is now part of the Ask search engine (www.Ask.com).

The main problem in text based searching the relevancy. Let take an example to understand it, user write a query such as "best book available on java". Now the search engine will do a text based search and it will find the number of occurrence of the word and renders the result back to the user which may consist of some 10000 of pages. But the question is that are they relevant or not. There could be pages containing information about coffee called java or few pages might be consisting information related to some they best but not based on java. Such pages when rendered back to the user are treated as irrelevant pages.

Now hits algorithm proposes an approach to overcome this. A page I is called authority for the query "best java book" if it contains valuable information on the subject. The official websites such as http://javarevisited.blogspot.in, http://www.javacodegeeks.com/, etc will be rendered and these are called authorities. These are the results that a particular user wants the search engine to return.

However, there is a second category of pages relevant to the process of finding the authoritative pages, called hubs. There role is to advertise the authoritative pages. They contain useful links towards the authoritative pages. In other words, hubs point the search engine in the "right direction".

K. Determine the Hubs and Authorities

Now we make some mathematical assumption to understand how Hits algorithm works. We associate to each page i two numbers; an authority weight ai, and a hub weight hi. We consider pages with a higher ai number as being better authorities, and pages with a higher hi number as being better hubs.

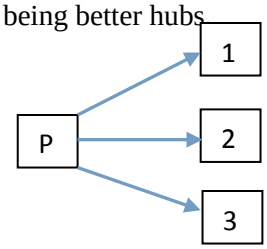


Figure 2.3 : Out bound link

a_p = the sum of hi for all nodes i pointing P

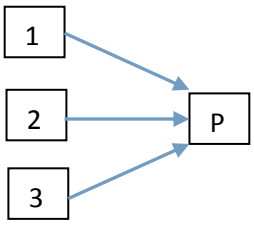


Figure 2.4 Inbound Link

h_p= the sum of a_i for all nodes i pointed to by P

Now we consider a very small example to calculate and understand the hits algorithm.

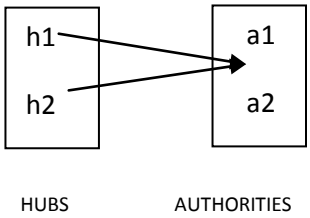
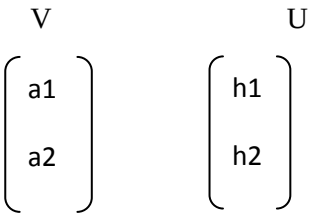


Figure 2.5: Hub and Nodes

Let A be the adjacency matrix of the graph S_Q and denote the authority weight vector by v and the hub weight by u, where



Now from the figure 2.5 we can clearly say that h3=1 and a2=2 because of the very simple reason is that h3 is calculated by number of different authorities it points to and hence in this case it only points to authority a2.The reason will remain same for hub h1=1.Now to calculate the authority of the page is that number of hubs points to that authority. In this case the hub h1 and h3 both points to the authority a2.Hence a2 have total value of 2.Below is the calculation done on basis of matrix. We have calculated the by considering that initial hub weight is 1.We have transposed the adjacent matrix and multiplied it with the hub to get the authority.

The adjacency matrix of the graph is A=

The transpose of A = A^t = $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$

Assume the initial hub weight vector is

$$U = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

We compute the authority weight vector by:

$$v = A^t.u = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix} . = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 2 \\ 2 \end{pmatrix}$$

Then the updated hub weight is:

$$u = A.v = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} . = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} \quad \begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix}$$

This already corresponds to our intuition that node 3 is the most authoritative, since it is the only one with incoming edges, and that nodes 1 and 2 are equally important hubs. If we repeat the process further, we will only obtain scalar multiples of the vector v and u computed at step 1.So the relative weights of the nodes remain same.

I. The Customization in Search Engine

This is our proposed theory in how to bring customization in search engine so that in minimum time the engine can give the result back to the user. But the question is how to achieve it, well Google has already done the customization in their search but the question remain can there be a better way to do the customization. The question could also be that why somebody wants customization when it already has been done. Well to answer that before Google was developed and way back to 1998 there were still many search engines existed who gave result relevant enough to the user. So if we could provide some better alternative and best result like Google has given then why not we should go for it?? That is underlying question.

J. The Proposed Methodology:

Web pages are classified into some categories. It is tough to determine the number of categories living under such a world. We have fixed out the number of categories in the classification. We search the web and update the database until it found some new category which is not present in their database. Once the number of category is decided then we determined which web page is under which category. Now we have to fix some keywords for all categories to decide that under which category a page will fall. There is a lot of issue for this categorization. One such issue is that a user writes a query such as “Virat Kohli and Anushka Sharma’s relationship status”. Yeah it seems little funny but there still millions of user exist who write such query every day. You can’t say that the user is naïve or he does not know proper English. This is our job to provide them related relevant information based on their query. Now the problem is both personalities’ falls under two different category

one in sports and other in celebrity or film industry. Now if such situation comes then that could be the possible solution. The first thing the search engine will do it look up its database where does the query gets matched. It may also happen that they were never been into a relationship and hence there is no news updated on it. If there is something, search engine will return that page back to the user. If it is not found, then it looks for the first word in the query that is "Virat" and second word is "Kohli".

Now we will start prioritize words from the index of the query. Of course only "Virat" would not give any result, but after appending the word "Kohli" it gives 100 of pages. Now there are two ways to give back to the user. The information of the first few pages for the first matched word is "Virat Kohli" and then next pages consist of word "Anushka Sharma". Another alternative is star ranking both the pages from "Virat Kohli" and "Anushka Sharma" and render the output based on the rank. There could be possible many more issues which really tough to identify. Now another task is to figure out the possible keywords by which we can do the web page classification. If we want to put some keywords for the topics of politics it could be "Narendra Modi, Mamata Banerjee, APJ Abdul Kalam" and so on. But the question is then the list of keywords would be endless. There are about hundreds of politician exist in different states and across the nation. The only possible solution is to do indexing of the keywords, got confused let me make you clear with an example. We use very simple concept of DBMS where a table will have four columns, the index keywords, category and the last one will a block pointer which acts as foreign key to some another table links are stored. We can fix the length of the keyword say 50, 60, or 100. If we get a new name for some politician then we check the last index of the category of politician. If the keyword field found to be filled and cannot accommodate the new keyword then we will create a new record and insert the keyword in along with index and category. Let us name this table as category table. Now when the user submit query related to politics instead of searching the whole millions of page it refers first to the category table to determines it category ant then the last column the address column refers to the main table to fetch those pages which are related to politics. Now say there are 1000 pages found and hence it dose ranking of only 1000 pages based on the text. Hence so millions of pages are not checked and hence lots of computational time was saves. This concept is same as multi-valued indexing in DBMS. To know more please refer to multi-valued indexing concepts. Now another thing could be possible done is at the search time user will given a dropdown to select the category and hence this make things easier.

K. THE IMPLEMENTATION OF OUR SEARCH ENGINE

The concept behind: We have successfully implemented page rank algorithm to design a search engine but in a very different approach. We have implemented the task using J2EE. The relevancy of pages was always checked in time to time and whenever a new page is found the first thing after authoritativeness done is to check the relevancy. Our aim is to build a search engine that gives few information but those are very relevant and important. Hence we have given the name of the search engine as **Relevant Important Page Search Engine(R I P S E).** The first thing the engine does it runs a text based search on the keywords on different pages. It then finds all the web pages those have those keywords in it. Then it crawl the web page and finds the entire link present in that web page. It stores the entire link in a data structure and recursively finds the entire link present in the newly found links. When a new link is encountered and it has got no existence in the database then the crawler crawls the web page finds the keywords by searching the meta tags, body contents, anchor text etc. Once the keyword is found we find the R-Factor of the web page. Then we implement the link based algorithm and find out pages which have got the highest number of rank. Then the result is returned back to the user. Now we will take small example to understand the concept behind. Suppose the user enters the query as "how to install tomcat server in eclipse". Now let say page p(A),p(B),p(C) were found the relevant

pages where the keywords matched were found. Now the crawler reads the page detects for further links. Each web page link is visited recursively and whenever a new link is encountered the crawlers does the same operation till it does not found any further links. Now after crawling the page p(A) let us say we got some new links (say 3 links) named them as p(A1), p(A2), p(A3). Now page p(A1) is visited and crawler reads the page to find links in it. If no link found it moves to the next page that is p(A2). Now to make things simple let us consider that after crawling the three links we got no further links in them. Then it move to the next page called p(B) and finds the links in them. Now say it finds a link called p(X) and p(Y). It the crawls the page p(X) and p(Y) and let say again to make things easier no further link were found. Now it moves to page P(A3) and after crawling it finds accidentally the same page called P(X).

3. CONCLUSION

In this paper, we have proposed a technique to design a search engine. Our primary goal is to create a search engine that can read human intentions. The search engine that focus on relevancy over information and importance over number of information. That is the reason we have introduced the R-factor and hence we gave the name of our search engine as PIPSE (Relevant Important Page Search Engine). We have proved how the introduction of the R-factor can change the ranking of the web pages. Future work can be done to calculate the R factor most efficiently and methodically. Future work is suggested in this perspective. Another factor where have adequately emphasized is for the customization in search engine.

References

[1] Kleinberg, Jon; "Authoritative Sources in a Hyperlinked Environment;" Proc. ACM-SIAM Symposium on Discrete Algorithms, 1998; pp. 668-677.

[2] Brin, Sergey; Page, Lawrence; "The Anatomy of a Large-Scale Hypertextual Web Search Engine;" 7th Int. WWW Conf. Proceedings, Brisbane, Australia; April 1998.

[3] Chakrabarti, S. et. al.,; "Mining the link structure of the World Wide Web;" IEEE Computer, 32(8), August 1999.

[4] Madria, Sanjay Kumar; 'Web Mining: A Bird's Eye View;" http://mandolin.cais.ntu.edu.sg/wise2002/slides.shtml; WISE 2002, Singapore.

[5] Xing, W.; Ghorbani, A.; "Weighted PageRank algorithm;" Proceedings of the Second Annual Conference on Communication Networks and Services Research, 19-21 May 2004; pp. 305 – 314.

[6] Wen-Xue Tao; Wan-Li Zuo;" Query-sensitive self-adaptable Web page ranking algorithm" Machine Learning and Cybernetics, 2003 International Conference on Volume 1, 2-5 Nov. 2003 Page(s):413 - 418 Vol.1

[7] Mukhopadhyay, Debajyoti; Giri, Debasis; Singh, Sanasam Ranbir; "An Approach to Confidence Based Page Ranking for User Oriented Web Search;" SIGMOD Record, Vol.32, No.2, June 2003; pp. 28-33.

[8] Brin, S.; Page, L. (1998). "The anatomy of a large-scale hypertextual Web search engine" (PDF). Computer Networks and ISDN Systems 30: 107–117.doi:10.1016/S0169-7552(98)00110-X. ISSN 0169-7552.