International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 04 | Apr 2022 www.irjet.net p-ISSN: 2395-0072

Calculating Rank of Web Documents Using Its Content and Link Analysis

Amit Kumar[1], Anshita Bhardwaj[2], Anshika Jain[3], Mr. Jagbeer Singh[4]

[1,2,3,4] - *Department of Computer Science and Engineering, Meerut Institute of Engineering and Technology, Meerut-250005, Uttar Pradesh, India*

Abstract - *On the World Wide Web (www), when a query is searched by the user over a search engine, ranking is the way through which the importance of web pages is measured by a search engine. In today's scenario, all the vital information is available online in the form of text documents. Various search engines are available for mining this available information, according to the user query, and giving appropriate and most relevant results to the user following his/her query. Search engines retrieve and show the documents according to their ranking. There are many search engines following page ranking for assignment of the weightage to the website's pages. In this paper, content-based matching is done along with the page ranking on hyperlink evaluation to display more accurate and relevant results following the user query.*

Key Words: Hyperlink evaluation, Ranking, Search engine, Search query, content-based.

1. INTRODUCTION

Nowadays, the Page-Rank method is mostly used in biblio-metrics[7], information networks, social analysis, and link prediction. It is also used for systems analysis of road networks and in Science, and neuroscience. The main factor is that it does not matter how long the query is, the answer will always come out in a particular order of links. Page-Rank seems very simple. But when a simple calculation is applied thousands or millions of times over the results can seem complicated. The main purpose of this paper is to provide an effective way to get the query result by using very simple code for clarity and understanding. The future work for starters can be, that we need to optimize our method by creating what our target audience wants to see. This will attract links better than anything else.

A search query is a string of words a user enters in the search box, and then the search engine gives the response within sub-seconds. A search engine is an online application that gets a query input from the user and based on the keywords or catchphrases received by the user, it fetches the results by online crawling [8] the websites with the help of crawlers or spiders, and then sorts them to make a list of hyperlinks corresponding to the matched documents.

In this paper, Along with the content-based matching, page ranking on hyperlink evaluation is done to display more accurate and relevant results following the user query. First, we have fetched out the links along with the content present inside the topmost text documents and pasted them inside a dictionary to evaluate a score to give the most relevant webpage, then the score is calculated for every document and a tagged score is assigned to each of them. After that, the highest score is found to get the best top pages reordered to improve user-fetched results on the search engine.

The responsive sequence of lists is also known as the Search Engine Result Page(SERP). The sequence of responses provided by search engines may consist of a mix of videos, images, articles, web pages, and many other types of files. The ranking of Web pages returned in response to a user query combines a measure of the relevance of the page to the query together with a query-independent measure of the quality of the page. The objective of this project is to reduce the uncertainty and un-usefulness of the web pages that come up at the top of the desired results by using both link and content analysis.

2. BACKGROUND HISTORY

The web pages shown at the top of the search results by the search engine are at times unwanted or useless for the user through certain practices. Mainly, web document retrieval has three types which are explained as:

2.1 Organic Search

Organic search is termed as the search methodology by which the search pages are retrieved through the search engine's algorithm. In the search engine's algorithmic test, web pages scoring exceptionally well are generally containing algorithms based upon factors such as quality and suitability of the content, specialization/expertise, authoritativeness, and trustworthiness of the website and its respective content writer on the given topic. Usually, the organic search results are the ones which are unpaid results appearing extensively over a search engine when the results page are popped up after the query gets searched by the user. For the sake of a relevant example, when user types "South Indian food" in any search engine, say, Google, there are all the unpaid results flashing which are all a part of the organic search. Commonly, people tend to view and open up the topmost results on the first page of all the search results. Each page of the search engine results, usually contains 10 organic listings[1,2], however, some results pages may have

different observations and have fewer organic listings in common.

2.2 Sponsored Search

A quite large number of companies or profit-making institutions find the method of showing up their services and business capabilities in the form of advertisements on the platform which is largely visited by interested users/customers. Therefore, a whole lot of popular search engines offer "sponsored results" to these organizations, who in turn pay these search engines for getting their products or services to appear above other search hits. This is often done in the form of auction/bidding among the companies, in which the bidder paying the largest amount or bidding, in this case, gets their hands on the top result as a part of their business goodwill and profit-making tactic. A sponsored search auction can also be called out as a keyword auction, which is largely an indispensable part of the business model being followed all over the globe mainly by the modern world web hosts. Here, the sponsored search results are the results referring to the results obtained largely through a search engine and are not at all extracted out from the main search algorithm, but rather from this business trick followed by interested companies in sponsored search result technique, also these results are very much the separate advertisements paid for by third parties[4,7]. A report submitted in 2018 from the house of European Commission showed that generally the consumers avoid these top results, as they have an expectation and mentality that the topmost results on a search engine page will be sponsored or baiting in the sense that they will be undesired, and thus quite less relevant to their information needs.

2.3 Popular web page Search

Many web document makers frame their web page in such a way that it contains a maximum number of linked pages to it so as to make the web document seemingly important and popular to the search engine algorithm. Also, the web pages linked are blank and totally unwanted at times. Here, The "about: blank" scenario comes into picture which is just a blank web page displayed when a user clicks on a highly attractive or free premium information which is also a kind of bait to the user. Here, the browser finds itself in a situation where it shows up the user an empty web page. Page Ranking is a method of measuring the significance of any available dynamic web pages. As per the Google, Page Rank mechanism by counting the occurrence number and its quality of links or relation to a page is to verify a rough approximation of how significant the website. According to a report over statistics from Statist, Net market share, and Stat Counter, the top 5 search engines worldwide in terms of market share are namely, Google, Bing, Yahoo, Baidu, and Yandex Google. With approximately over 70% of the search market share, Google is undoubtedly the most popular search engine. Additionally, Google captures almost 85% of mobile traffic. But, here the case is that the web page does not get served to you from an external source, so it isn't harmful to our computer. However, in most of the cases, the cause behind showing up a blank page can be the malware which can make the browser open a blank page.

3. WORK FLOW OF PROPOSED METHODOLOGY

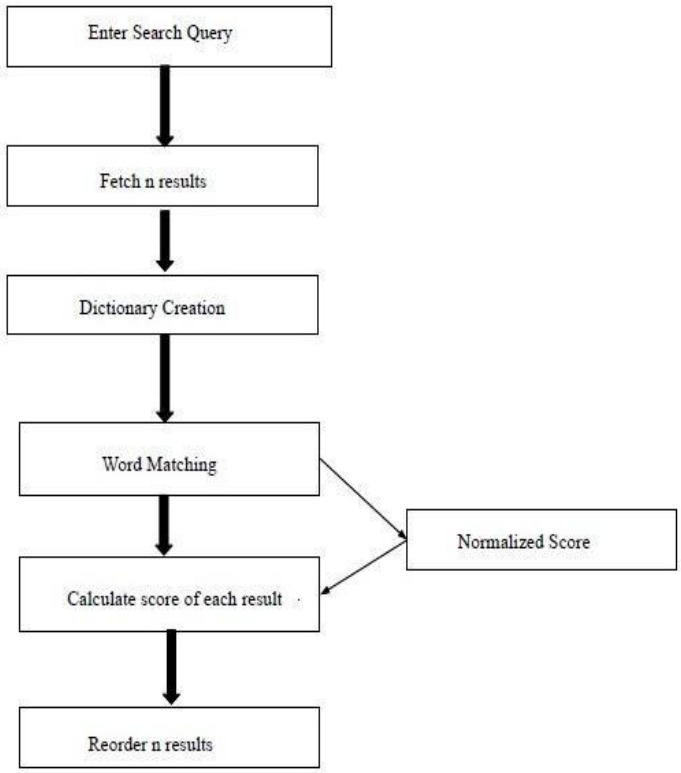

fig 1: Work flow of proposed methodology

4. Ranking

The scoring of searched query results is done by a component known as query processor which is entitled with computing scores of web pages through the effective use of a ranking procedure that is highly dependent upon the retrieval model. Each and every ranking procedure innately depends on such a model. Highly used form of finding the score in every web document is as mentioned below [5]

$$\sum_j t_j * w_j$$

Here, t_j = j^{th} weight of query term,

w_j = j^{th} weight of document term

And, the summation is done for each term present inside the dictionary of the collection for the repository.

5. MODULE DESCRIPTION

a. Fetch n results

First search a query or pattern (e.g.: Happy) and discover the first 3 links. And after that, pick up the content inside those links and explore the searching pattern or query whether it is on screen or not.

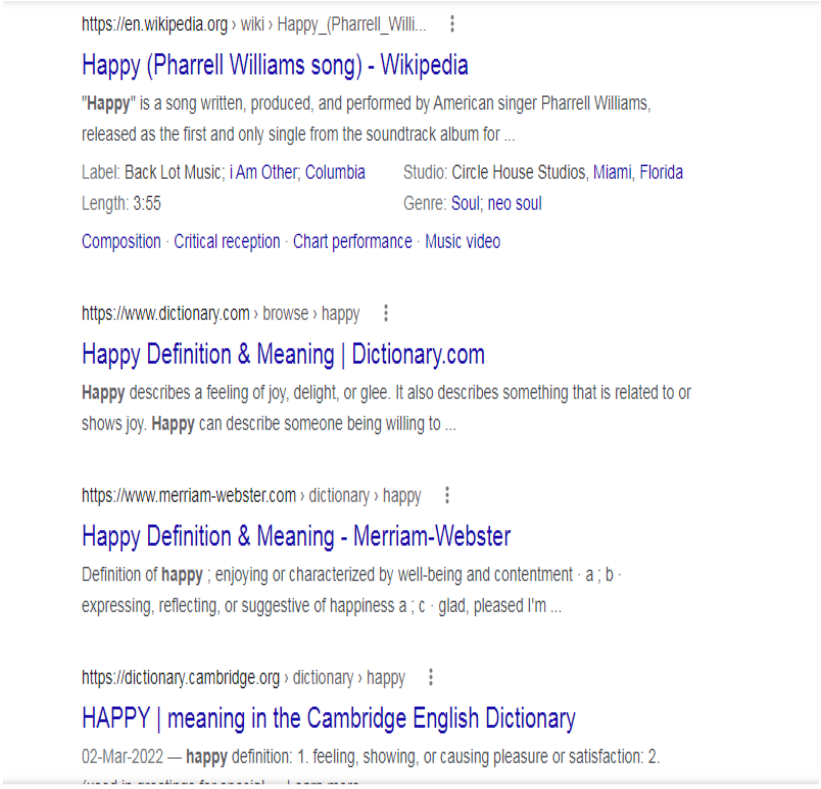

fig 2: Sample search result

b. Dictionary Creation

1. Download the word net module of python for dictionary usage.
2. Import that module to the main python program by using the following commands -from serp-api import Google Search from nltk. corpus import wordnet.

c. Word Matching

```
import dictionary modules
{       integer total=0
          q = input query from user
          initialize score[] array
        total=0,sc=0
          for int i=0 to n:
                  if(query[i] in dictionary[words]):
                          sc=sc+1
                          total=total+sc
                  elif(query[i]                    in
dictionary[synonyms])::
                          score=score+0.5
                          total=total+score
                  score[i]=sc;
      average_score=total/n
      if(score[i]>threshold_value)
       {
            score[i]=score[i]/average_score;
       }}
```

d. Sample Code for Execution

```
q=input("Enter the querry: ")       # Taking querry
input from user
params = {
 "engine": "google",
 "q": q,
 "google_domain": "google.com",
 "api_key":
"94a571b4a4482b0d19564027ebdc0bd28a6383f6
3a076a5829978637e65ddbdd"
}                          #parameters for google search
search = GoogleSearch(params)      #A varible that
will store the search results
results = search.get_dict()        #A variable which
will store search in dictionary form
org_re=results['organic_results']    # we need only
organic results
old={}                     # An empty dictionary
j=0                        # A counter variable
for i in org_re:
 old[j]={'rank' : i['position'] ,'title' : i['title'] ,'link' :
i['link'] ,'description':i["snippet"],'score':j}   #store
only desired parameters
 j += 1
print("\n----------The results are----------
---\n") #printing the search results
for i in old:
 print(old[i]['title'])
 print(old[i]['link'])
 print('\n')
synonyms = []              # A variable for storing
the synonyms of the searched querry
q=q.split(' ')             #for setences we split the
word for their sysnonyms''l
for i in q:
 for syn in wordnet.synsets(i):
  for l in syn.lemmas():
   synonyms.append(l.name())
synonyms=set(synonyms)
print()          #remove duplicate words by making
the set
total=0
for i in old:
 score=old[i]['score']
 url=old[i]['link']
```

```
 total=total+score
 try:
  request_result=requests.get( url )
  soup = bs4.BeautifulSoup(request_result.text,"html.parser") # Creating soup from the fetched request
  all_text=soup.get_text()
  score+=all_text.count(q)
  for j in synonyms:
   score+=all_text.count(j)/2
  old[i]['score']=score
 except:
  pass
new={}                          # An empty dictionary
l=len(old)                      # no of search results recived
sc_l=[]
for i in old:
 sc=old[i]['score']
 if sc in sc_l:
  sc+=.1
 sc_l.append(sc)
 average_score=total/(len(old))
 new[sc]={'title': old[i]['title'],'link' : old[i]['link'],'description':old[i]["description"],'rank': old[i]['rank']}
print("\n\n\n----------The Updated results are----------\n") #printing the Updated search results
for i in sorted(new,reverse=True):
 print("prev rank: ",new[i]['rank'])
 print(new[i]['title'])
 print(new[i]['link'])
 print("Score: ",old[i]['score'])
 print('\n')
print("Average score is ",average_score)
```

5.1 Demonstration of the algorithm with the help of an example

Step 1: Fetch the links related to entered query

```
Enter the querry: great
https://serpapi.com/search

------------------The results are----------------------

GREAT | meaning in the Cambridge English Dictionary
https://dictionary.cambridge.org/dictionary/english/great


Great Definition & Meaning - Merriam-Webster
https://www.merriam-webster.com/dictionary/great


GREAT - Translation in Indonesian - bab.la
https://en.bab.la/dictionary/english-indonesian/great


231 Synonyms & Antonyms for GREAT | Thesaurus.com
https://www.thesaurus.com/browse/great


Great Definition & Meaning | Dictionary.com
https://www.dictionary.com/browse/great


Great definition and meaning | Collins English Dictionary
https://www.collinsdictionary.com/dictionary/english/great
```

fig 3: Fetched links of the entered query (here, query is 'great')

Step 2: Pick up the synonyms from word net dictionary

```
Synonyms are:
slap-up

neat

groovy

expectant

keen

bang-up

large

enceinte

smashing

swell

outstanding

dandy

not_bad

corking
```

fig 4: Synonyms from wordnet dictionary (here, query is 'great')

Step 3: Calculate the score and n-score of each query term

fig 5: Calculated score of the query (here, query is 'great')

6. EXPERIMENTAL RESULT ANALYSIS

Query-Searched	Output		Results of Score		Number of links Reordered
	Query Term	Number of appearances	N-Score	Score	
What is Great	Great	2,37,00,000	4.5	9	6
OS Dead Lock	OS Dead lock	57,50,000	3.5	7	4
What is Happiness	Happiness	2,18,00,000	4	8	2
Advantages of Java	Advantages java	1,23,00,000	4	8	0
What is Database Management system	Database Management system	2,23,00,000	3.5	7	1
Essay on India	Essay India	4,77,00,000	3.5	7	7
Downloading Python IDE	Downloading Python IDE	2,37,00,000	3.0	6	3
Hello	Hello	3,26,00,000	3.0	6	7
Voting Results in UP	Voting Results UP	2,14,00,000	4.0	8	5
Election 2022	Election 2022	2,37,00,000	2.0	4	2
Uttar Pradesh	Uttar Pradesh	2,11,00,000	2.5	5	3

Code Link and Queries relationship

```
xpoints = np.array([1, 2, 3, 4])
ypoints = np.array([6, 4, 2, 0])
x2points = np.array([5,6])
y2points = np.array([1,7])
plt.plot(xpoints, ypoints,marker= 'o')
plt.plot(x2points,y2points,'o')
font1 = {'family':'serif','color':'blue','size':20}
font2 = {'family':'serif','color':'darkred','size':15}
plt.title("Graphical Result Analysis", fontdict = font1)
plt.xlabel("Queries", fontdict = font2)
plt.ylabel("Links Reordered", fontdict = font2)
plt.show()
```

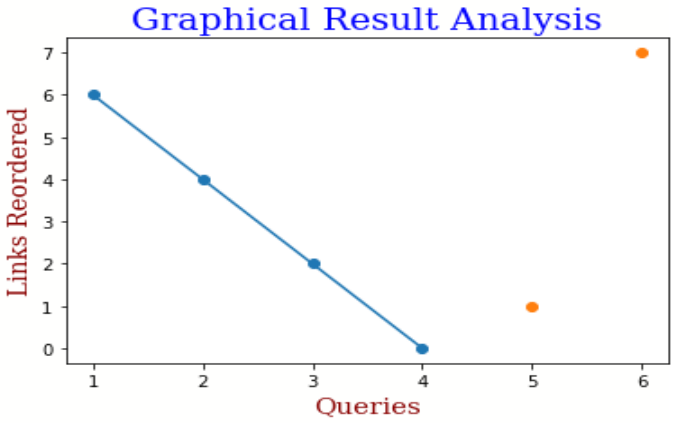

Chart -1: Links Reordered vs Queries

Code for N-Square and Queries relationship

```
xpoints = np.array([1, 3, 5])
ypoints = np.array([4.5, 4, 3.5])
x2points = np.array([2,4,6])
y2points = np.array([3.5,4,3.5])
plt.plot(xpoints, ypoints,marker='o')
plt.plot(x2points, y2points,'o')
font = {'family':'serif','color':'darkred','size':15}
plt.xlabel("Queries", fontdict = font2)
plt.ylabel("N-Score", fontdict = font2)
plt.show()
```

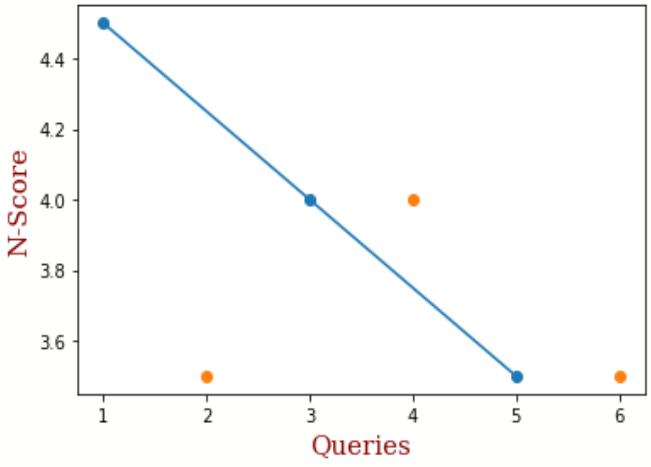

Chart -2: N-Score vs Queries

Code for Queries and Score relationship

```
xpoints = np.array([1,3,5])
ypoints = np.array([9,8,7])
x2points = np.array([2,4,6])
y2points = np.array([7,8,7])
plt.plot(xpoints, ypoints,marker='o')
plt.plot(x2points, y2points,'o')
font = {'family':'serif','color':'darkred','size':15}
plt.xlabel("Queries", fontdict = font2)
plt.ylabel("Score", fontdict = font2)
plt.show()
```

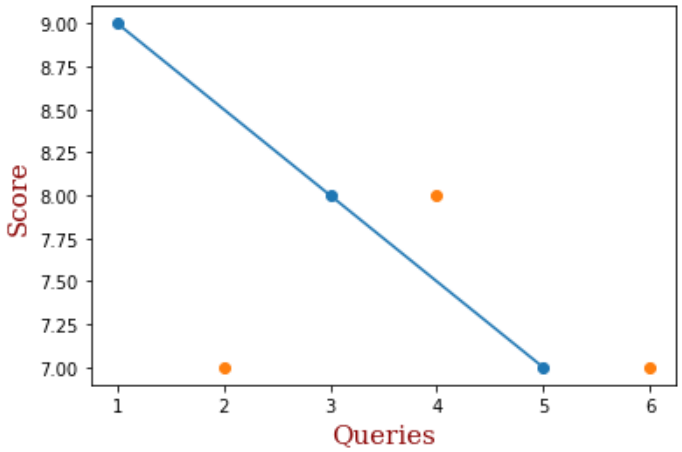

Chart -3: Score vs Queries

Code for Comparison and Queries relationship

```
x1 = np.array([1, 2, 3, 4])
y1 = np.array([6, 4, 2, 0])
x2 = np.array([1, 2, 3])
y2 = np.array([4.5, 4, 3.5])
x3=np.array([1,2,3])
y3=np.array([9,8,7])
plt.plot(x1, y1, x2, y2, x3, y3)
font = {'family':'serif','color':'darkred','size':15}
plt.xlabel("Queries", fontdict = font)
```

```
plt.ylabel("Comparison", fontdict = font)

plt.show()
```

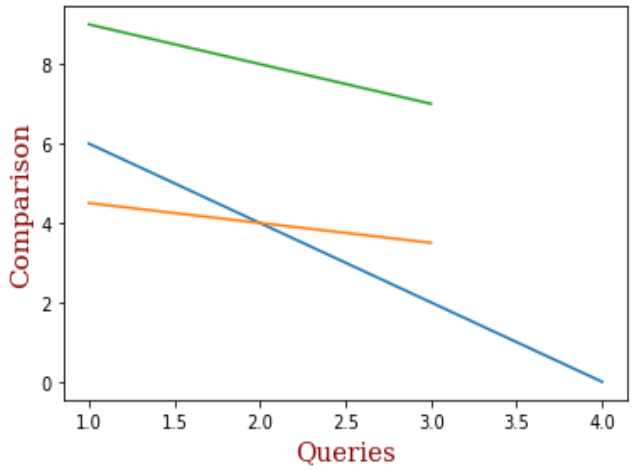

Chart -4: Comparison of all parameters vs Queries

7. CONCLUSION AND FUTURE WORK

According to the Graph, after the application of the ranking algorithm, the page's ranking is changing by 4% accurately. Therefore, the above algorithm is effective for bringing about the top results of search engines in the user's desired order. Page Ranking is a method of measuring the significance of any available dynamic web pages. As per the Google, Page Rank mechanism by counting the occurrence number and its quality of links or relation to a page is to verify a rough approximation of how significant the website.

REFERENCES

[1] Swati Jain, Mukesh Rawat (2020) Efficiency measures for ranked pages by Markov Chain Principle, International Journal of Information Technology(2020), Volume 13, Issue 6.

[2] Nandnee Jain, Upendra Dwivedi (2015) Ranking web pages based on user interaction time'', International Conference on Advances in Computer Engineering and Applications, IEEE Xplore, pp. 35–41, March 19–20

[3] Vojnovic M, Cruise J, Gunawardena D, Marbach P (2009) Ranking and suggesting popular items. IEEE Trans Knowl Data Eng 21(8):1133–1146

[4] Nandnee Jain, Upendra Dwivedi (2015) Ranking web pages based on user interaction time'', International Conference on Advances in Computer Engineering and Applications, IEEE Xplore, pp. 35–41, March 19–20

[5] Bruce Croft W, Metzler D, Strohman T (2015) Search engines information retrieval in practice. Pearson Education, London, pp 25–26

[6] Ishii H, Tempo R (2014) The page rank problem, multiagent consensus, and web aggregation: a systems and control view-point. IEEE Control Syst Mag 34(3):34–53

[7] Chakrabarti S, Dom B, Gibson D, Kleinberg J, Kumar R,Raghavan P, Rajagopalan S, Tomkins A (1999) Mining the link structure of the world wide web. IEEE Computer Soc Press 32(8):60–67

BIOGRAPHIES

Mr. Jagbeer Singh is the faculty assistant professor at Meerut Institute of Engineering and Technology, Meerut, U.P, India.

Anshika Jain is a final year graduate in Computer Science and Engineering at Meerut Institute of Engineering and Technology, Meerut, U.P, India.

Anshita Bhardwaj is a final year graduate in Computer Science and Engineering at Meerut Institute of Engineering and Technology, Meerut, U.P, India.

Amit Kumar is a final year graduate in Computer Science and Engineering at Meerut Institute of Engineering and Technology, Meerut, U.P, India.