LINK-BASED WEB RANKING ALGORITHMS BASED ON WEIGHTED GRAPH USING PROBABILISTIC APPROACH

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Software Engineering

Submitted By
Preet Kamal
Roll No. 801031023

Under the supervision of:
Mr.Ravinder Kumar
Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004
June 2012

LINK-BASED WEB RANKING ALGORITHMS BASED ON WEIGHTED GRAPH USING PROBABILISTIC APPROACH

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Software Engineering

Submitted By
Preet Kamal
Roll No. 801031023

Under the supervision of:
Mr.Ravinder Kumar
Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004
June 2012

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, **"Link-Based Web Ranking Algorithms based on weighted graph using probabilistic approach"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Mr. Ravinder Kumar* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Signature
(Preet Kamal)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Ravinder Kumar)
Assistant Professor,
CSED

Countersigned by

(Dr. Maninder Singh)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

i

ACKNOWLEDGEMENT

It is a great pleasure for me to acknowledge the guidance, assistance and help I have received from Mr. Ravinder Kumar. I am thankful for his continual support, encouragement, and invaluable suggestions. He not only provided me help whenever needed, but also the resources required to complete this thesis report on time.

I am also thankful to Dr. Maninder Singh for his kind help and cooperation.

I would also like to thank all the staff members of Computer Science and Engineering Department for providing me all the facilities required for the completion of my thesis work.

I would like to say thanks for support of my classmates. I want to express my appreciation to every person who contributed with either inspirational or actual work to this thesis.

I am highly grateful to my family for the inspiration and ever encouraging moral support, which enabled me to pursue my studies.

Preet Kamal

ABSTRACT

The web today plays an important role in the cultural, educational and commercial life of millions of the users. With the huge amount of information available on the web, users typically rely on the web search engines in order to get the most desired and relevant information. A web search engine's task is to find the most relevant content on the web regarding the query asked by the user. Even if they allow giving the user relevant pages for any search topic, the numbers of results returned are often too big for the user to be carefully explored. Also, the needs of the different users vary as what seems to be important for one user may be completely irrelevant for the other user As most of the users examine the first few pages so the key for user satisfaction is to give the desired results in the first few pages thus there is a necessity to have an efficient ranking algorithm. Therefore the role of ranking algorithms is crucial i.e. select the pages that are most likely be able to satisfy the user's needs and also bring those results to the top positions. Over the past decade many ranking algorithms have been proposed but there is a little research performed on the measuring the effectiveness of these algorithms. In this research a new method for making the adjacency matrix of the web graph for ranking algorithms is suggested. The rank of ranking algorithms like those of PageRank, HITS, SALSA and Norm (P) algorithms are calculated. From the rank values various performance measures like Mean Reciprocal Rank, Mean Average Precision and Normalized Discounted Cumulative Gain values are calculated and their efficiency is compared with the present scenario of considering the outlink of a particular webpage. In this we also suggest how to calculate relevancy from the document rank.

Key words: Hyperlinks, ranking, Page rank algorithm, HITS algorithm, SALSA algorithm, MRR, MAP and NDCG value

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The web today plays an important role in the cultural, educational and commercial life of millions of the users. With the huge amount of information available on the web, users typically rely on the web search engines in order to get the most desired and relevant information. In a web search engine, due to the dimensions of the current web and the needs of the users, its role becomes very critical. The expectations from search engines are very high. Users often ask vague questions from the search engines and expect concise and organized response. Users type "committee" and expect the search engine to retrieve results for "committee". In short users expect the computer to supply the information they want to search for instead for the words they type.

It may sound strange, users are not interested in keeping tabs on knowing how a particular search engine work. Users are always interested in taking the desired information. Once users get their desired result they log off from the system. So it is concluded that the users are the supreme authority to decide whether the retrieved information is according to his or her requirements. In the field of information retrieval, this is known as relevance i.e. judging how well the retrieved results match the query posted by the user. Information retrieval is the field of study consists of searching the documents, looking for information within these documents and for collecting the metadata about the documents.

Recent studies [1] estimated the existence of around 11.5 billion web pages on the web. These days, it is common for simple search queries to return thousands or millions of results. Most of the users do not have the time to go through all of these results one by one and look for the one in which the user is interested. A recent study [2,3] shows that the user rarely go beyond the first few pages of results. Therefore, the role of ranking algorithms becomes very critical. Each search engine has a unique ranking algorithm that parses its database of webpages to determine relevant responses to the queries of the user.

1.1 WEB GRAPH

The web graph describes the directed links between various webpages of the World Wide Web. A graph consists of several vertices and edges. In a directed graph, edges are directed lines or arcs. The web graph is a directed graph, whose vertices represent the pages of WWW and a directed edge connects page X to page Y if there is a hyperlink on page X, referring to page Y.

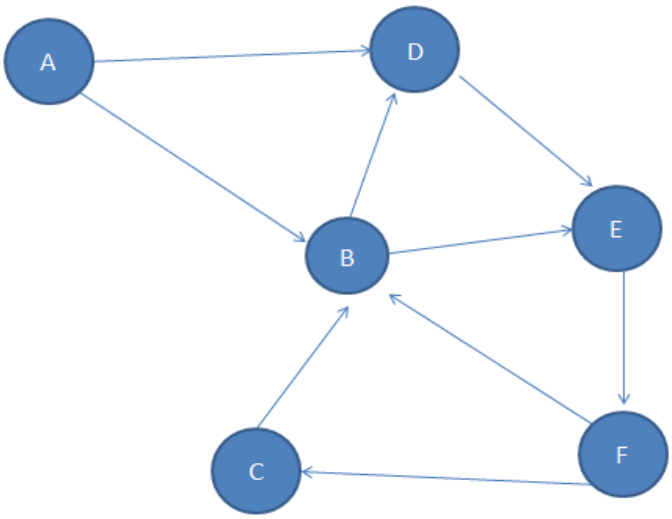


Figure 1.1 the webgraph

A sample small web graph.In this example we have six pages labeled A-F. Page B has in-degree 3 and out-degree 2.

The connectivity of graphs is defined is terms of the paths: two nodes are linked by a path if we can follow a sequence of edges from one to another. A graph is connected if every pair of nodes present the graph are linked by a path. A path from a node A to node B is the sequence of nodes beginning with A and ending with B in a directed graph. It must hold the property that each of the consecutive pair of nodes forming the sequence is connected by an edge which is pointing in the forward direction. Thus we can conclude that a directed graph is strongly connected if there is a path from every node to every other node. Thus the graph represented in *fig 1* is not strongly connected.

We say that a strongly connected component (SCC) in a directed graph is a subset of the nodes such that: (i) every node in the subset has a path to every other; and (ii) the subset is not part of some larger set with the property that every node can reach every other.

The first large scale study of the webgraph was done by Broder et al.[4] and it was found that the webgraph has a giant component which further contains three different components of approximately same size :

i. the giant SCC(strongly connected component): which is made up of single strongly connected component
ii. IN set: nodes that can reach the giant SCC but cannot be reached from it
iii. OUT set: nodes that can be reached from the giant SCC but cannot reach it

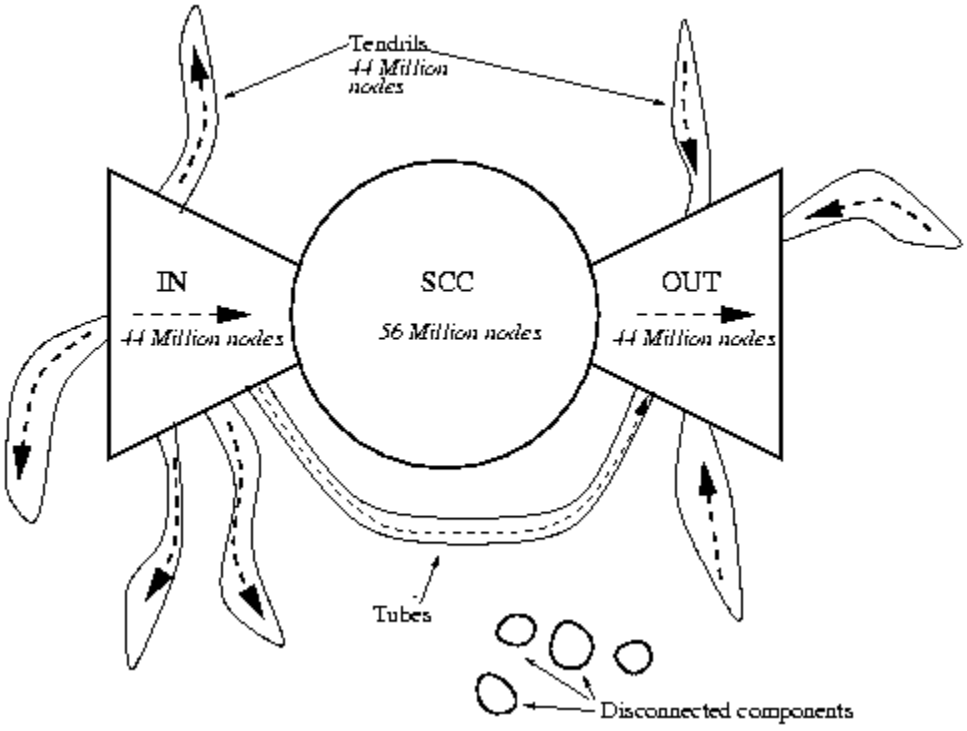


Figure 1.2 The bow tie structure of the web

Fig 2 represents the macroscopic structure of the web[4]. Understanding the inner details of the web graph is important to improve the performance of algorithms that rely on the link structure of the web.

1.2 HYPERLINK ANALYSIS ON THE WEB

Hyperlink analysis makes two following assumptions:

- If two pages are connected then they might be on the same topic.
- Author of page1 may recommend page2 by sending a hyperlink from page1 to page2.

There are basically two uses of hyperlink analysis namely crawling and ranking which are discussed in detail below:

1.2.1 Web Crawling

It is basically the process of collecting web pages. In the crawling process documents are found out by the web search engine which starts from a set of source web pages and the same process is continued until a predetermined number of pages have been found or no new set of pages are found. The order in which the web pages are crawled is fixed by the crawler of the search engine, for instance the crawler of a search engine may crawl web pages based on quality.

1.2.2 Ranking

After receiving a query from user, the URL's of the documents are returned by the search engines to the user in the decreasing order of relevance of these documents, hence this process is known as ranking.

Two types of graphs are used for ranking the web pages:

- **Co-citation graph:** In an undirected co-citation graph, nodes A and B are connected by an undirected edge if and only if there exists a third page C hyperlinking to both A and B. it may be said that A and B are co- cited by C.

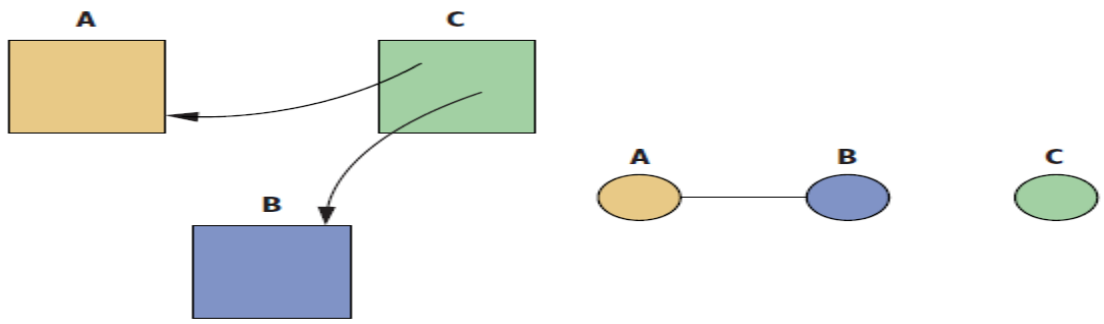


Figure 1.3 Co- citation Graph depicting existence of hyperlink between nodes.[5]

- **Link Graph:** Link graph follows a simple approach representing each web page by a node and stating that there exists a hyperlink between two web pages only if the nodes are connected by a direct edge.

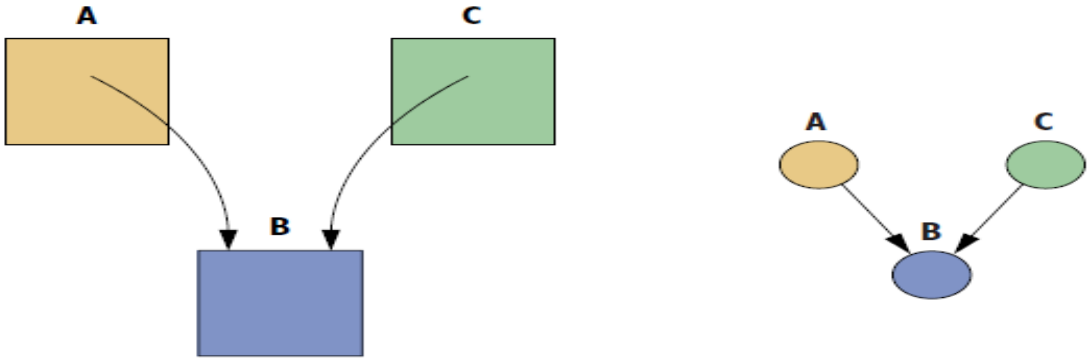


Figure 1. 4 Link Graph depicting existence of hyperlink between nodes.[5]

Connectivity- based ranking is further divided into two fields:

- Query independent schemes
- Query dependent schemes

1.2.2.1 Query independent schemes

In this type of scheme a score is assigned to each page based on some criterion prior to receiving a query from the user and this score is used to rank the page on arrival of a query. One of the simplest example might be the page ending with most number of edges in the graph is the graph having most hyperlink request and hence has the highest quality.

1.2.2.2 Query dependent schemes

In this type of scheme the ranking algorithm assigns ranks to a particular page depending on the query received. A query specific neighbourhood graph is built and a hyperlink analysis is performed on this graph.

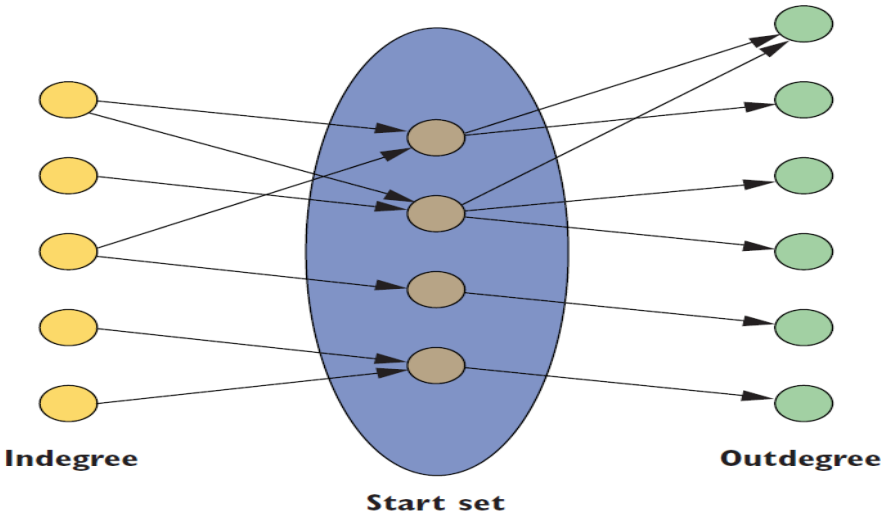


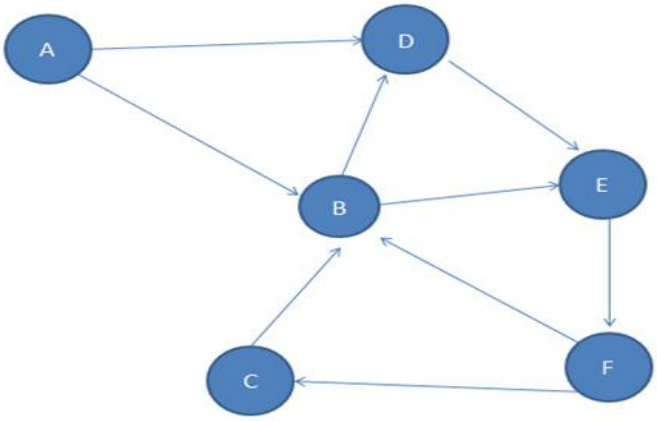
Figure 1.5 the query based neighbourhood graph [5]

1.3 MATHEMATICAL REPRESENTATION

Let P denote the resulting set of nodes of the webgraph, and let n be the size of the set P. Let G = (P,E) denote the underlying graph, where each directed edge connects two nodes if there is a link exists between them. The input of any link analysis algorithm is the adjacency matrix W of the graph G where W[i, j] = 1 if there is a link from node i to node j and if there is not any link between them then W[i,j]=0. The output of the algorithms is an n-dimensional vector a, where a_i is the authority weight of node i in the graph. These weights are used to rank the pages.

For the given webpages Wi and Wj the adjacency matrix M = ($m_{i,j}$) defined as

$$m_{i,j} = \begin{cases} 1 & if\ Wi \rightarrow Wj \\ 0 & otherwise \end{cases}$$



$$\begin{Bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{Bmatrix}$$

Figure 1.6 Graph and corresponding mathematical representation

1.4 EXISTING LINK BASED ALGORITHMS

1.4.1 Indegree algorithm

A simple heuristic that can be thought as the predecessor of all Link Analysis Ranking algorithms is to rank the pages according to their popularity [6]. The popularity of a page can be measured by the number of pages that link to that given page. We refer to this algorithm as the INDEGREE algorithm, since it ranks pages according to their in-degree in the graph G. This simple heuristic was applied by several search engines in the early days of Web search.

1.4.2 HITS algorithm

HITS (Hyperlink-Induced Topic Search) are a query-based algorithm. From the user query, the HITS algorithm first creates a neighborhood graph for the query. The neighborhood graph contains top 200 matched web results retrieved from a content-based web search engine. It contains all the pages of the 200 web pages linked to and web pages that linked to these 200 top pages. After this an iterative calculation is performed on the values of authority and hub. For each webpage p, the authority and hub values are calculated. The authority value of webpage p is the sum of hub scores of all the webpages that points to p, the hub value of page p is the sum of authority scores of all the webpages

that p points to (Fig.7). Iteration proceeded on the neighborhood graph until the values converged. Then, iterative calculations were performed on the value of authority and value of hub. For each page p, the authority and hub values are computed as follows:

$$a(i) = \sum_{(j,i)\in E} h(j)$$

$$h(i) = \sum_{(i,j)\in E} a(j)$$

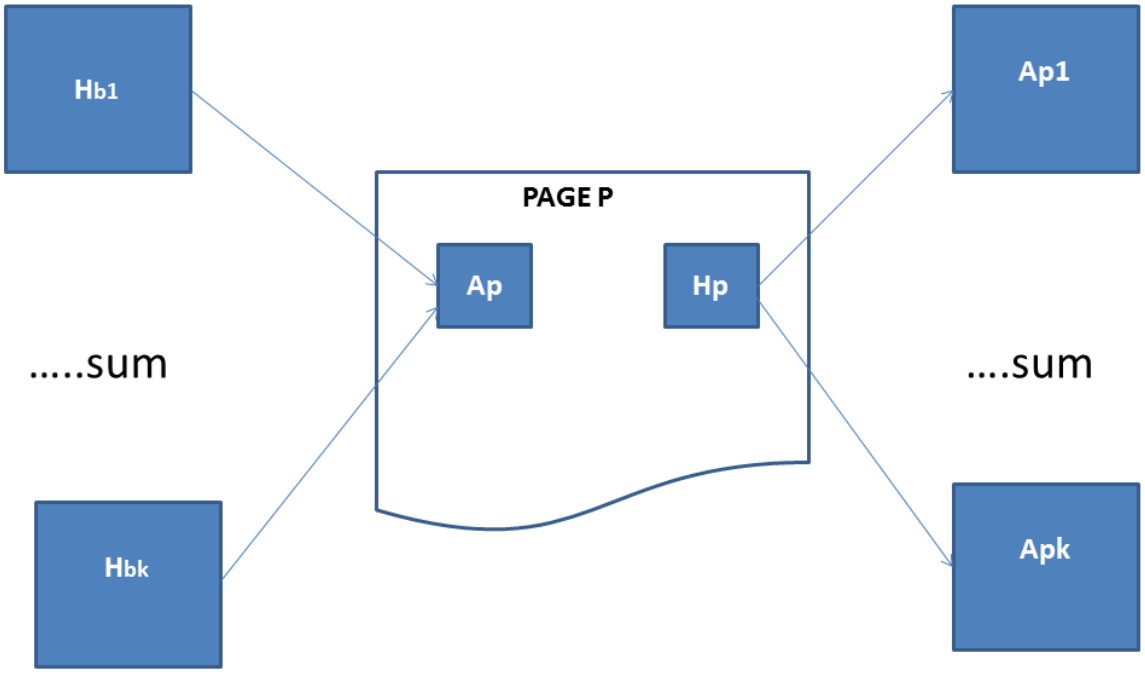


Figure 1.7 HITS Algorithm

1.4.3 PageRank Algorithm

Page Rank is a query independent algorithm, which is based on the connectivity structure of the web pages. It is used by Google search engine. The Page Rank value of a page is weighted by each hyperlink to the page proportionally to the quality of the page containing the hyperlinks; i.e., the Page Rank value of a page will spread evenly to all the pages it points to. The Page Rank R (p) of a page p can be defined as the probability that the surfer is at the page on a given time step:

$$\mathrm{PR}(P) = (1-d) + d(\frac{PR(T1)}{C(T1)} + \cdots \ldots . + \frac{PR(Tn)}{C(Tn)})$$

Where PR (p) is the Page Rank of a page p

PR (T1) is the Page Rank of a page T1

C (T1) is the number of outgoing links from the page T1

d is a damping factor in the range 0 < d < 1, usually set to 0.85

The Page Rank of a web page is therefore calculated as a sum of the Page Ranks of all pages linking to it (its incoming links), divided by the number of links on each of those pages (its outgoing links).

1.4.4 SALSA algorithm

The SALSA algorithm combines the ideas both from page rank and HITS algorithm. In this algorithm, a random walk on the bipartite hubs and authorities graph alternatively between hubs and authorities is performed. When on an authority side of the bipartite graph at a node, the algorithm selects one of the incoming links uniformly at random and moves to a hub node on the hub side. When at node on the hub side the algorithm selects one of the outgoing links uniformly at random and moves to an authority.

1. The hub matrix defined as:

$$h_{i,j} = \sum_{\{k|(i_h,k_a),(j_h,k_a \in G\}} \frac{1}{\deg(i_h)} \frac{1}{\deg(k_a)}$$

2. The authority matrix defined as:

$$a_{i,j} = \sum_{\{k|(k_h,i_a),(k_h,j_a \in G\}} \frac{1}{\deg(i_a)} \frac{1}{\deg(k_h)}$$

1.5 EVALUATION OF SEARCH ENGINES

There are several different measures that are proposed to measure the performance of classical information retrieval systems but most of them extend to evaluate web search engines. Most of the web users have a tendency to give more preference to a particular performance issues as compared to users of the traditional information retrieval systems. For example response time has the most priority as compared to other performance issues. A basic model from traditional retrieval systems recognizes a three way trade-off between the speed of information retrieval, precision, and recall as shown in the figure below:

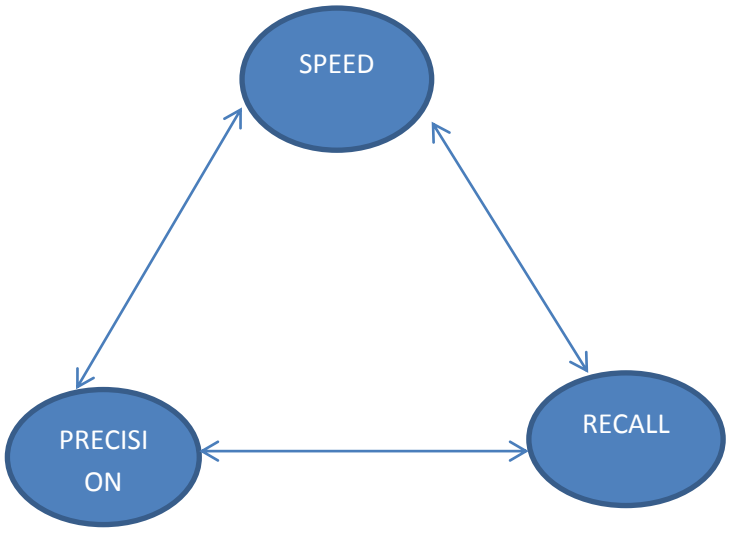


Figure 1.8 Three way trade-off in search engine performance [7]

The trade-off becomes very difficult to balance with the increase in number of documents and users. Precision is defined as the ratio of relevant documents to number of retrieved documents.

$$Precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|}$$

Recall is defined as the proportion of relevant documents that are retrieved.

$$Recall = \frac{|\{\ relevant\ documents\} \cap \{retrieved\ \ documents\}|}{|\{\ relevant\ documents\}|}$$

Most of the internet users are not interested in the traditional measure of precision as the precision of the results shown in the first page of the retrieved documents. There is little scope of measuring the recall rate for each search query. A web user is more concerned about retrieving and able to identify highly relevant pages. The probability that a non-relevant document is retrieved by the query is defined as Fall-out.

$$Fall - out = \frac{|\{non - relevant\ documents\} \cap \{retrieved\ \ documents\}|}{|\{non - relevant\ documents\}|}$$

The weighted Harmonic mean of precision and recall is given by F measure as

$$F = \frac{2.precision\ .recall}{(precision + recall)}$$

By calculating precision and recall at every position in the ranked sequence of documents, a precision–recall curve can be plotted by plotting precision *p(r)* as a function of recall. Average Precision computes the average value *p (r)* over the interval from r =0 to r =1.

$$Average\ Precision = \int_0^1 p(r)dr$$

1.6 STRUCTURE OF THE THESIS

In Chapter 2 the Literature Survey of existing Algorithms for Ranking is presented.

In Chapter 3 discussion the Proposed Solution for Problem Statement

In Chapter 4 comparison of the present scenario with the proposed solution is represented.

In Chapter 5 Conclusion and future scope is outlined.

CHAPTER 2

LITERATURE SURVEY

Ranking is one of the most integral components of any information retrieval system. In the case of Web search, due to the size of the Web and the special nature of the Web users, the role of ranking becomes critical. It is common for Web search queries to have thousands or millions of results in response to query asked by the web user. On the other hand Web users do not have the time and patience to go through them to find the ones they are interested in. Therefore, it is crucial for the ranking function to output the desired results within the top few pages.

2.1 PAGE RANK

PageRank is one of the most important ranking techniques used in today's search engines i.e. Google. PageRank is a simple, robust and reliable way to measure the importance of web pages, but it is also computationally advantageous with respect to other ranking techniques in that it is query independent, and content independent [8]. The Google theory says if Page A links to Page B, then Page A is saying that Page B is an important page. PageRank also factors in the importance of the links pointing to a page. If a page has important links pointing to it, then its links to other pages also become important.

2.1.1 Calculation of PageRank

PageRank is a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. It can be calculated for collections of documents of any size. The PageRank computations require several passes, called "iterations", through the collection to adjust approximate PageRank values to more closely reflect the theoretical true value.

2.1.2 Simplified PageRank Algorithm:

In this it is assumed that a particular web graph consists of only few numbers of hyperlinks. Initially all the page rank calculations are performed on the small set of data. It uses the following calculations [8]

$$pr(A) = (1-d) + d \sum_{s \in I_A} \frac{pr(s)}{\deg(s)}$$

Where pr(s) is the PageRank of the vertex s

I_A is the in-neighbour set of the vertex A

deg(s) is the out-degree of vertex s

$\sum_{s \in I_A} \frac{pr(s)}{\deg(s)}$ *is the summation of PageRank of all the webpages pointed to given webpage.*

Algorithm 1 PageRank algorithm(G,s,k)

1: d← 0.85

2: n ← number of vertices in G

3: **for** i= 0 to n **do**

4: pr[i]← s

5: **end for**

6: **for** j=0 to k **do**

7: **for all** pr[i] **do**

8: pr_{in} ← sum of all incoming normalised PageRanks

9: pr[i] ← (1-d) + d(pr_{in})

10: **end for**

11: **end for**

12: avg ← sum(pr[i])/n

Let us see this using an example. In this we have a hypothetical set of pages titled A,B,C and They link to each other as shown below

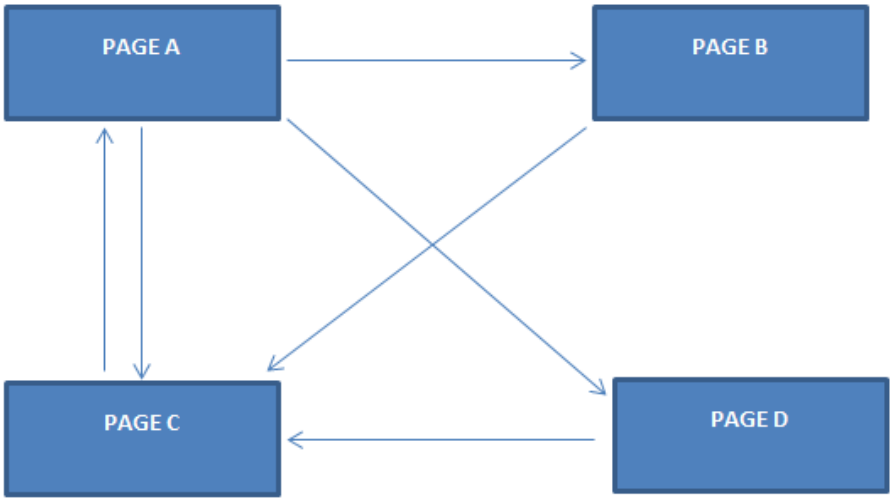


Figure 2.1 Webgraph of 4 hypothetical pages

The rules for necessary calculation to obtain the PageRank for each page are:

1. We take a 0.85 * a page's PageRank and divide it by the number of links on the page.

2. We add that amount on to a new total for each page it's being passed to.

3. We add 0.15 to each of those totals.

Here it started with zero so 0 * 0.85 is always 0. So each page gets just 0.15 + 0. that is each page now has a PageRank of 0.15.

A links to pages B, C and D. Page A's PageRank is 0.15 so it will add 0.85 * 0.15 = 0.1275 to the new PageRank scores of the pages it links to. There are three of them so they each get 0.0425.

Page B links to page C. Page B's PageRank is 0.15 so it will add 0.85 * 0.15 = 0.1275 to the new PageRank score of the pages it links to. Since it only links to page C, page C will get it all.

Page C links to Page A, all 0.1275 passes to page A.
Page D links to Page C. Again all 0.1275 passes to page C.

The new totals for each page them become:
Page A=0.15 (base) + 0.1275 (from Page C) = 0.2775
Page B=0.15 (base) + 0.0425 (from Page A) = 0.1925
Page C= 0.15 (base) + 0.0425 (from Page A) + 0.1275 (from Page B) + 0.1275 (from Page D) = 0.4475
Page D=0.15 (base) + 0.0425 (from Page A) = 0.1925

It can be easily seen that Page C is probably the most important page in the system. These calculations are carried on until the value for each page no longer changes this is called convergence. In practice, Google probably doesn't wait for this convergence, but instead run a number of iterations of the calculation which is likely to give them fairly accurate values.

2.1.3 Damping Factor

The web is modelled using a directed graph and the assumption is that it is strongly connected i.e. every page can be reached by following links from any other page. In practice, this is not the case and as a result a damping factor D is incorporated in the PageRank equation. The damping factor is described by L. Page and S. Brin [8] as "a vector over the Web pages which is used as a source of rank to make up for the rank sinks such as cycles with no outedges".The value selected for D can be any value in the range 0 <D < 1, although as the damping factor goes to 1, the rank of all important nodes goes to 0. A value of 0.85 suggested by L. Page and S. Brin is more commonly used. The original paper from Page & Brin [8] states that this is usually set somewhere between 0.85 and 0.9. Adjusting its value will not only effect the convergence time of the iterative calculation, but also the qualitative properties of the results. A higher damping factor will result in a higher level of accuracy in the PageRank value, but at the cost of requiring more iterations of the algorithm to reach convergence.

2.1.4 Convergence

It means that whatever values we start at, after doing the calculation a number of times we will get the same final values and these values will no longer change even if we perform further iterations of the calculation. These final values are known as limiting values. Once the limiting values have been calculated, there is no need to expend processing power on calculating the PageRank.

2.1.5 Modified Page rank algorithm

This formula uses the model of a random surfer who gets bored after clicking on several links and then switches to a random page. The PageRank value of a given page reflects the chance that the random surfer will come to that page by clicking on a link. So it can be understood as a Markov chain in which the states are pages, and the transitions are all equally probable and are the links between the pages. If a page has no links to other pages, it becomes a sink and thus terminates the random surfing process. If the random surfer arrives at a sink page, he picks another URL at random and continues surfing again.

So the equation becomes

$$\mathrm{PR}(p_i) = \frac{q}{N} + (1-q) \sum_{p_j \in L(p_i)} PR(p_j)$$

Where p_1,p_2,p_3 …..p_n are the pages under consideration(p_i) is the set of pages that link to pi and N is the total number of pages.

The PageRank values are the entries of the dominant eigenvector of the modified adjacency matrix. This makes PageRank a particularly elegant metric: the eigenvector is

$$R = \begin{bmatrix} PageRank(p_1) \\ PageRank(p_2) \\ \vdots \\ PageRank(p_N) \end{bmatrix}$$

Where R is the solution of the equation

$$R = \begin{bmatrix} \frac{q}{N} \\ \frac{q}{N} \\ \vdots \\ \frac{q}{N} \end{bmatrix} + (1-q) \begin{bmatrix} l(p_1,p_1) & l(p_1,p_2) & \cdots & l(p_1,p_N) \\ l(p_2,p_1) & \ddots & \vdots & \vdots \\ \vdots & \cdots & l(p_i,p_j) & \vdots \\ l(p_N,p_1) & \cdots & \cdots & l(p_N,p_N) \end{bmatrix} R$$

Where the adjacency function $l(p_i,p_j)$ is 0 if page p_i does not link to page p_j and normalised such that for each i

$$\sum_{j=1}^{N} l(p_i,p_j) = 1$$

2.1.6 Advantages of Page Rank Algorithm:

- Due to its query independent nature it is able to answer to queries a lot faster because it has the pre computed page ranks.
- There is no spamming problem in Page Rank algorithm because it uses global measures instead of the local neighbourhood graphs so the owner of a web page cannot increase his rank by increasing inlinks to his page.

2.1.7 Disadvantages of Page Rank Algorithm:

- Since Page Rank is query-independent, it cannot by itself distinguish between pages that are authoritative in general and pages that are authoritative on the query topic.
- There exists the problem of topic drift which says that the search engine may link pages that are irrelevant to the subject of the query made by the user.

2.2HITS Algorithm

HITS stands for Hyper Text Induced Topic Search. It was developed by Kleinberg and he introduced the concept of hubs to prove his point that it is not necessary that good authorities point to other good authorities [9]. Every page is considered as having two identities, hub identity and the authority identity.

- A hub is considered to be a good hub or a quality page if it points to good authorities.

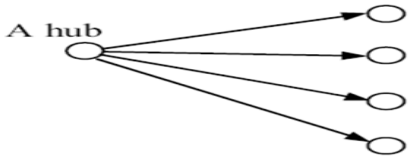


Figure 2.2 Representing a HUB [10]

- Authority is good if it is pointed to by good hubs.

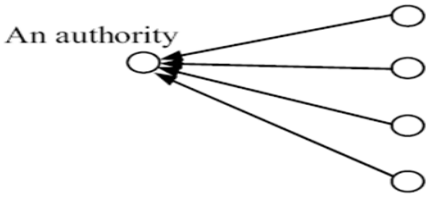


Figure 2.3 Representing an Authority [10]

In order to quantify the quality of a page as a hub and an authority, Kleinberg associated with every page a hub and an authority weight.

Whenever a query is run by the user the search engine provides the user with the relevant pages. The main objective is to keep the set of these relevant pages really small. The PageRank algorithm computes the page ranks of the entire web graph. On the contrary HITS algorithm tries to discover authority and hub pages in a given sub graph.

When the relevant pages are grouped into a subset S, there is always a problem of this set becoming really large and one can end up with over a million pages returned by the search engine. The main reason behind this problem is that mainly the companies' web sites today have a graphical outlook without the display of text or also a word like "laptop" may appear in the web sites of every company.

To overcome the problem mentioned above HITS algorithm is implemented which focus on the following properties of subset S of relevant pages [11]:

1. S is relatively small.
2. S is rich in relevant pages

3. S contains most of the strongest authorities

2.2.1 Root Set in HITS algorithm

HITS algorithm is started with the root set of pages R that were obtained using the text-based search engine. This set is then inflated by adding the pages pointed to, or that point to, any page in the root set. An additional parameter "d" is introduced to make sure that a single page in the root set is allowed to bring at most "d" pages pointing to it in the root set.

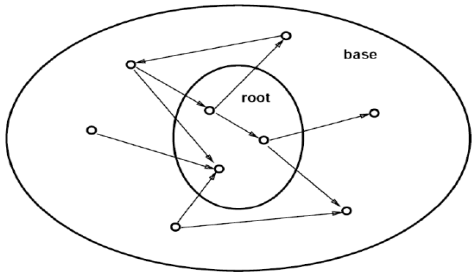


Figure 2.4 Representing the expansion of root set[11].

2.2.2 Working of HITS algorithm

Each page p is assigned with two weights, namely

- Non-negative authority weight denoted by a_p.
- Non-negative hub weight denoted by h_p.

After each of these weights are normalized so that the sum of each of them comes out to be equal to unity. Higher the value of "a" better is the authority and greater the value of "h" better is the hub.

In the HITS algorithm for a given query q a set of t highest ranked pages are selected. These pages are known as root set. From this base set S is constructed by including any page pointed to by a page and any page points to a page. For a given page i in S an authority score a(i) and hub score h(i) is assigned[4].

$$a(i) = \sum_{(j,i)\in E} h(j)$$

$$h(i) = \sum_{(i,j)\in E} a(j)$$

Algorithm 2 HITS algorithm(G,k)

1: G← a collection of n linked pages

2: k← a natural number

3: L← adjacency matrix of the graph

4: **Initialize** $a_0 = h_0 = (1,1,\ldots 1)$

5: HITS –**Iterate (G)**

Let z denote the vector$(1,1,1,1,\ldots 1) \in R^n$

$a_0 \leftarrow h_0 \leftarrow z$

k← 1

6: **Repeat**

Apply the I operation to (a_{k-1}, h_{k-1}) as follows

$$a_k \leftarrow L^T L a_{k-1}$$

Apply the O operation to (a_{k-1}, h_{k-1}) as follows

$$h_k \leftarrow L L^T h_{k-1}$$

$$a_k \leftarrow \frac{a_k}{\|a_k\|}$$ //normalization

$$h_k \leftarrow \frac{h_k}{\|h_k\|}$$ //normalization

$$k \leftarrow k + 1$$

7: **until** $|a_k - a_{k-1}| < E_a$ and $|h_k - h_{k-1}| < E_h$

8: **return** a_k and h_k

Mathematically saying there is a reinforcing relationship between hubs and the authorities stated below:

- If p points to many pages with large a-values, then it should receive a large h-value.

$$a_p = \sum_{q:(q,p)} h_q$$

- If p is pointed to by many pages with large h-values, then it should receive a large a-value.

$$h_p = \sum_{q:(q,p)} a_q$$

2.2.3 Advantages of HITS Algorithm

- HITS algorithm provides with the dual rankings which provides the most authoritative documents to the user related to a query and also the hub documents concerned with it.
- The size of the matrices formed in the implementation phase is really small as compared to the total number of documents on the web.[12]

2.2.4 Disadvantages of HITS Algorithm

- Because of the query dependence neighbourhood graph has to be built every time a query is encountered.
- HITS algorithm is also susceptible to spamming. By adding links to and from his webpage, a user can slightly influence the authority and hub scores of his page which can seriously influence the ranking of the pages retuned to the user. It can be exploited commercially by website owners to improve the ranking of their web page.

- There is also presence of topic drift according to which a very off topic page linked by the algorithm while answering the query.[12]

2.3 SALSA ALGORITHM

Stochastic Approach for Link-Structure Analysis or SALSA developed by Lempel and Moran, combines the random walk idea of PageRank with the hub/authority idea of HITS[1]. It is variant of the Kleinberg's HITS algorithm with the sole difference that SALSA weighs the entries on the basis of their in and out degrees rather than using rather than using the straight adjacency matrix as in HITS . The algorithm relies upon the theory of Markov chains, and is based on the stochastic properties of random walks performed on our collection of pages. It uses a bipartite graph explained below:

2.3.1 Bipartite graph:

In the field of mathematics a bipartite graph is one whose vertices can be represented in two disjoints sets such none of the vertices in the same set are adjacent to each other. A bipartite graph G is a graph where the set of vertices can be divided into sets V1 and V2 such that each edge is incident on one vertex in V1 and one vertex in V2

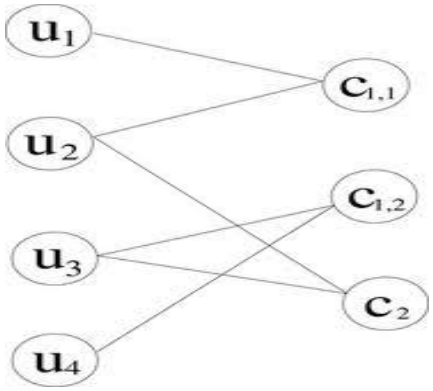


Figure 2.5 Bipartite graph[13]

2.3.2 Working Of Algorithm

Given a graph a bipartite undirected graph undirected graph G = {Va, Vh, E} is built where[14],

- Vh = (the hub side of G).
- Va = (the authority side of G).
- E = Set of directed edges.

And m is less than or equal to 2n. (m and n are the number of nodes in H and G respectively)

The SALSA algorithm implements a random walk on the bipartite undirected graph by transferring between the Hub side and Authority side of the graph. The procedure can be described in the following steps.

Step 1: Random walk starts at some authority node selected randomly and keeps on transferring between the two sides.

Step 2: While on the authority side of the graph the algorithm randomly selects one of the incoming links and moves to a hub node on the hub side of the graph.

Step 3: While on the hub side of the graph algorithm randomly selects one of the outgoing links and moves to an authority node on the authority side of the graph.

Hence we obtain the two Markov chains authority chain and the hub chain [14] and their relative stochastic matrices:

1. The hub matrix defined as:

$$h_{i,j} = \sum_{\{k|(i_h,k_a),(j_h,k_a \in G\}} \frac{1}{\deg(i_h)} \frac{1}{\deg(k_a)}$$

2. The authority matrix defined as:

$$a_{i,j} = \sum_{\{k|(k_h,i_a),(k_h,j_a \in G\}} \frac{1}{\deg(i_a)} \frac{1}{\deg(k_h)}$$

Algorithm 3 SALSA Algorithm(G,s,k)

1: B ← neighbourhood graph

2: $$B = \bigcup_{u \in R} \{u\} \cup \ S_m[\{v \in V : (u,v) \in E\}] \cup \ S_n[\{\in V : (u,v) \in E\}]$$

3: $B^A = \{ u \in B : in(u) > 0\}$

4: **For** all u ∈ B **do**

$$A(u) = \begin{cases} \frac{1}{|\ B^A\ |} & if\ u \in\ B^A \\ 0 & otherwise \end{cases}$$

5: **Repeat** until A converges:

For all $u \in\ B^A$ **do**

$$A'(u) = \sum_{(v,u) \in N} \ \sum_{(v,w) \in N} \frac{A(w)}{out(v)in(w)}$$

For all $u \in\ B^A$ **do** A(u) = $A'(u)$

2.3.3 Example of SALSA Algorithm:

In the example given below the working of the algorithm is explained [15]. Given is a graph and a bipartite graph G is obtained from the same.

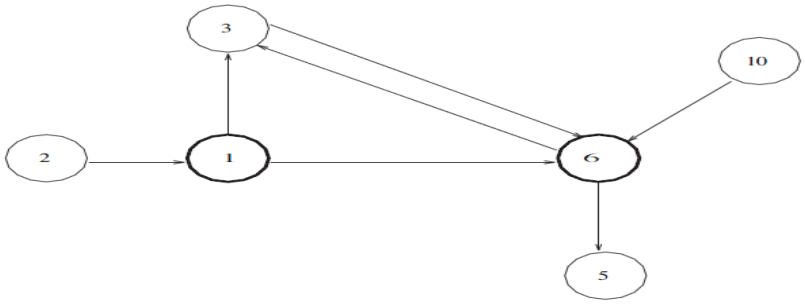


Figure 2.6 Neighbourhood graph[15]

Bipartite graph has three attributes (Vh, Va, E) such that:

- Vh is the set of hub nodes (all nodes with outdegree > 0)
- Va is the set of authority nodes (all nodes with indegree > 0)
- E is the set of directed edges

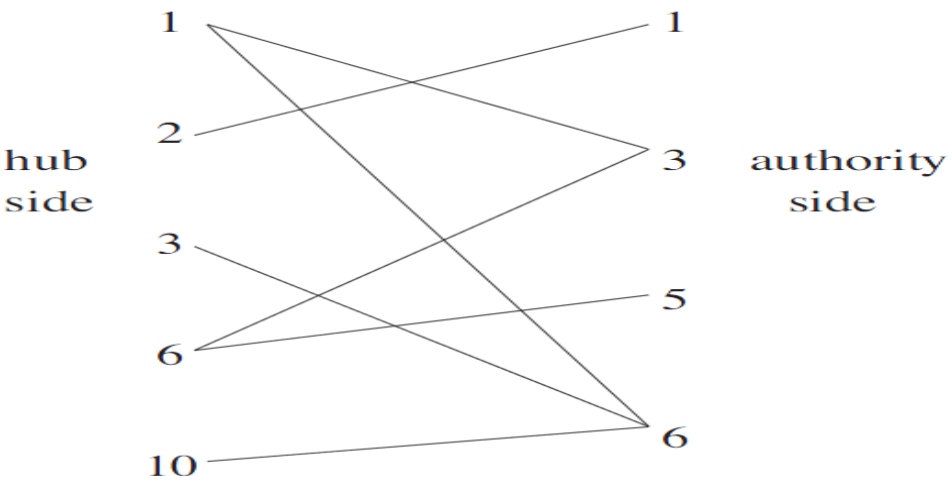


Figure 2.7 Hubs and authorities in bipartite graph[15]

2.3.4 Advantages of SALSA

- It is not affected by topic drift problem, which means that when a query is sent by the user, then the algorithm does not reply with pages that are off topic or do not have any relevance with the query.
- SALSA is also less prone to spamming problem avoiding the manipulation of page ranks by the web page owners.

2.3.5 Disadvantages of SALSA

- It handles spamming better than HITS. Spamming means adding links to and from his webpage, a user can slightly influence the authority and hub scores of his page which can seriously influence the ranking of the pages retuned to the user but not near as good as PageRank

- Due to its query dependence nature neighborhood graph and bipartite graphs are to made for each query thus making it slower as compared to the query independent algorithms.

2.4 NORM(P) ALGORITHM

It works on the principle that small authority weights should contribute less to computation of the hub weights[16] Norm is a function which assigns a positive length to all the vectors in the given vector space. These algorithms belong to the family of additive online learning algorithms. These work on the principle of preferential treatment of the authority weights. It can be implemented by using a norm or an operator. By this we will be able to use the fact that lower authority weights contribute less to the hub weight. The simplest approach is to scale the weights. Now the question is how to choose the scaling factors. The most common solution to this question is to use the authority weight to determine the scaling factor. As higher authority weights are significant in the calculation of hub weight so hub weight of the given node i is set to be the p-norm of the vector of the authority weights of all the nodes pointed to by the given node i.

Algorithm 4 Norm(P) algorithm(G,s,k)

1: **Repeat until convergence**

2: **O operation:** hubs compute the p-norm of the authority weight vector

$$h_i = (\sum_{j:i \to j} a_j^p)^{\frac{1}{p}} = \|F(i)\|_p$$

3: **I operation:** authorities collect the hub weights

$$a_i = \sum_{j:j \to i} h_j$$

4: **Normalise** weights under some norm

CHAPTER 3

PROBLEM STATEMENT AND PROPOSED SOLUTION

3.1 PROBLEM DEFINITION

In the previous chapters a literature survey of the existing webpage ranking algorithms for the Web information retrieval is presented. Information retrieval process begins with the user has some information needs. These information needs are presented in the form of a query using the query representation. On the other hand all the documents are represented by the document representation. Information retrieval system matches the two representations to determine the documents that satisfy the user's information needs. The problem with the matching is both query and document representations are uncertain. Given a set of documents D = { d_1, d_2, …., d_n } and the query q in what order of the subset of the relevant documents Dr ={d_{r1},d_{r2},……$d_{r\ n}$} should be returned to the user i.e. we want the best document to be at the rank 1 , second best to be at rank 2 and so on.

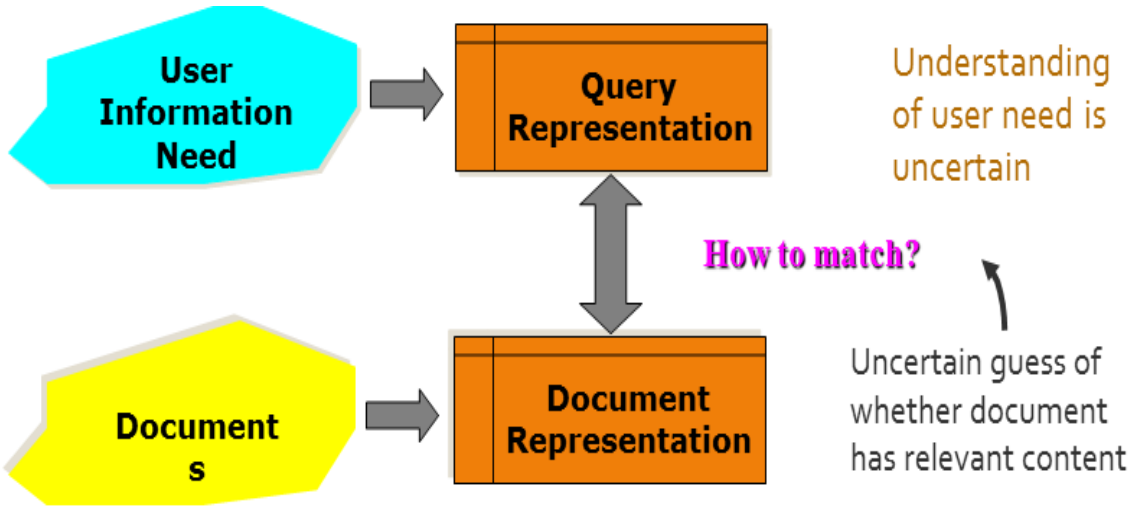


Figure 3.1 Relationship between Query representation and document representation

With the present scenario of making the adjacency matrix from the webgraph the results for the Mean Reciprocal Rank,Mean Average Precision and Normalised Discounted Cumulative Gain of various webpage ranking algorithms are presented as follows

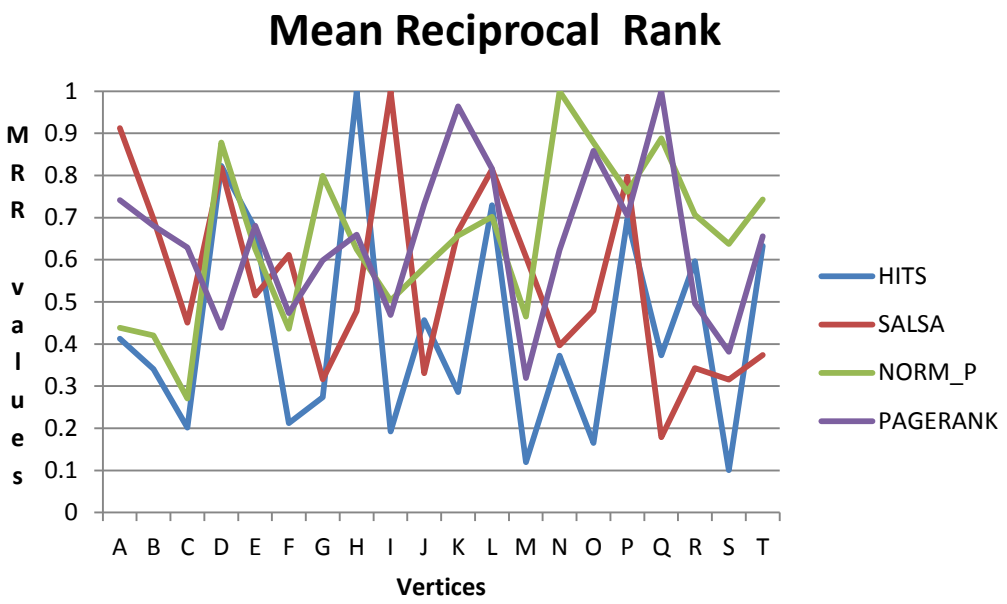


Figure 3.2 Mean Reciprocal Rank for the ranking algorithms

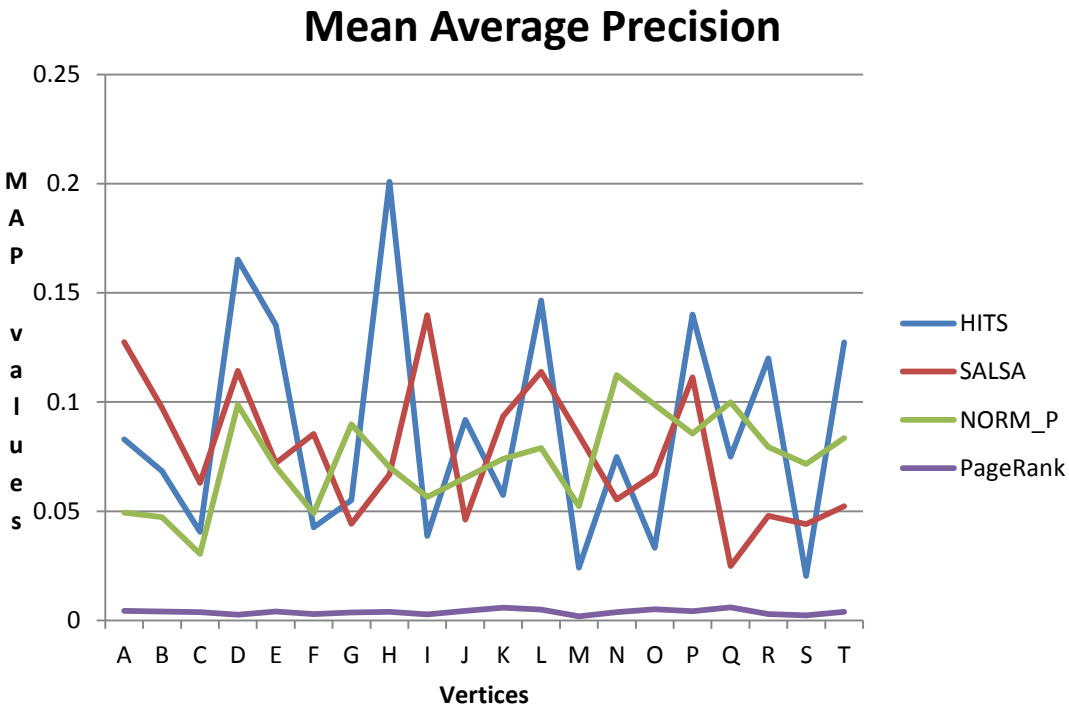


Figure 3.3 Mean Average Precision for the ranking algorithms

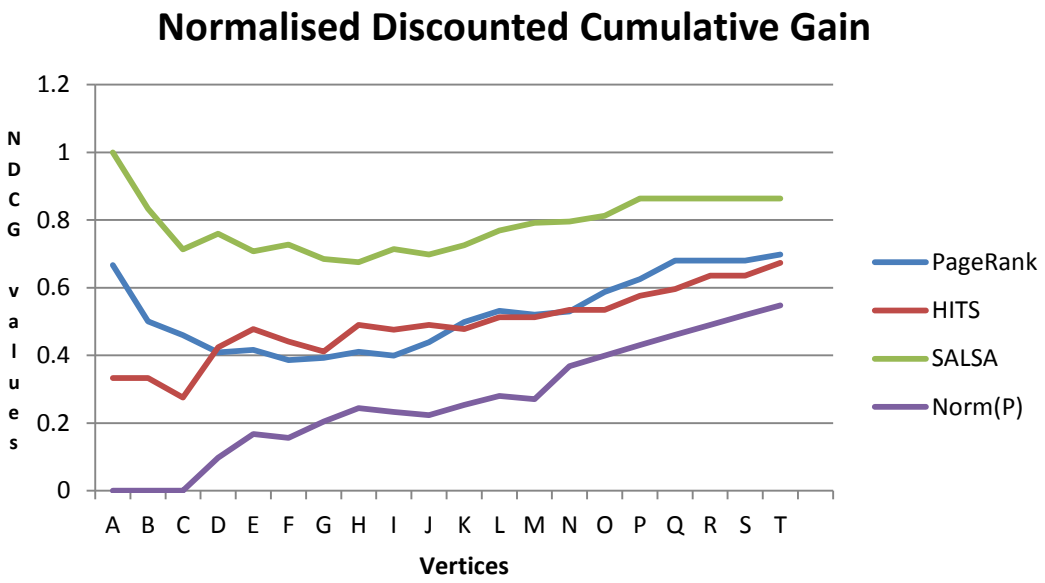


Figure 3.4 Normalised Discounted Cumulative Gain for the ranking algorithms

3.2 PROPOSED SOLUTION

To deal with Uncertainty, Probabilities provide a principled foundation for uncertain reasoning. Probabilistic methods are one of the oldest but one of the currently most favoured and suitable approach for the given problem. The basic idea is to utilise the principle of probability to make the adjacency matrix for the given webgraph.

3.2.1 Probability

Probability is defined as the phenomena of calculating the certainty of an event occurrence and when this certainty is measured in numeric value then this is known as probability. The probability of any event always lies between 0 and 1. It means that it cannot be less than zero and cannot be greater than one. It deals with random variables and experiments but these are always well defined within the given problem. The probability of an event A is represented by P[A].

3.2.2 Conditional Probability

Whenever the occurrence of any event is restricted due to any other event or condition then this situation is defined as Conditional Probability and in this the chances of occurrences of an event get limited. The conditional Probability of an event E_1 given event E_2 is the probability of that event E_1 occurs, given that event E_2 occurs. It is denoted by $P[E_1][E_2]$ and can be defined as :

$$\mathrm{P}\, E_1|E_2 = \frac{\Pr[E_1 \cap E_2]}{\Pr[E_2]}$$

$P[E_1|E_2]$ is defined as the probability of event E_1 when the sample space S is restricted to event E_2.It is also noted that $P[E_2] \neq 0$.

3.2.3 Bayes Theorem

Bayes Theorem relates one conditional probability (e.g., the probability of a hypothesis H given an observation E) with its inverse (the probability of an observation given a hypothesis). Bayes Theorem states that

$$P[H|E] = \frac{P[E|H].P[H]}{P[E]}$$

The theorem directly follows from the definition of conditional probability

$$P[H|E] = \frac{P[H \cap E]}{P[E]} = \frac{P[E|H].P[H]}{P[E]}$$

In this study we have used the probability of visiting the webpage Wj from the webpage Wi in the adjacency matrix instead of 0 and 1.In the webpage Wi all the links are analysed and hence the probability of visiting the given hyperlink is calculated. If there are 4 hyperlinks on the webpage Wi then the adjacency matrix is defined as

$$m_{i,j} = \begin{cases} \dfrac{1}{number\ of\ outlinks\ on\ Wi} & if\ Wi \rightarrow Wj \\ 0 \quad otherwise \end{cases}$$

CHAPTER 4

EXPERIMENTAL RESULTS

In this chapter description of the experiments that are carried out with the present scenario of considering the links between the two pages only is compared with the proposed scenario of considering all the links on a given page while constructing the adjacency matrix on the basis of Mean Average Precision, Mean Reciprocal Rank and Normalised Discounted Cumulative Gain on the algorithms like those of Page Rank,HITS ,SALSA and Norm(P) algorithms.

4.1 EXPERIMENTAL SET UP

Experiments were performed on a hyperlink graph of 20 vertices. For the purpose of this study a webgraph having 20,000 nodes which are interconnected have been selected. These 20,000 nodes are subdivided into 20 nodes which are named from {A to T} in order to manage and calculate their respective values effectively. On this webgraph PageRank, HITS, SALSA and Norm(P) algorithms are applied. When the rank of these algorithms are calculated the performance measures like those of MAP,MRR and NDCG values are determined.

Determining the retrieval results from the webpages scores and human judgement is not very common and are the subject of research on the field of information retrieval. An efficient performance measure should take into account the user satisfaction keeping the fact that they will not like to dwell deep to obtain the desired results.

4.2 INPUT DATA SET AND MEASUREMENT

For the purpose of this study a webgraph having 20,000 nodes which are interconnected have been selected. These 20,000 nodes are subdivided into 20 nodes which are named from {A to T} in order to manage and calculate their respective values effectively. On this webgraph PageRank, HITS, SALSA and Norm(P) algorithms are applied. When the rank of these algorithms are calculated the performance measures like those of MAP,MRR and NDCG values are determined.

Determining the retrieval results from the webpages scores and human judgement is not very common and are the subject of research on the field of information retrieval. An efficient performance measure should take into account the user satisfaction keeping the fact that they will not like to dwell deep to obtain the desired results.

4.2.1 The neighbourhood Graph of the result set

The algorithms discussed in this paper are based on topical endorsements. By topical endorsement we mean that the hyperlink on a given page u on a given topic endorses the authority of another page v with respect to given topic. Due to this topical endorsement it is economical to perform the link analysis on the hyperlinks that are endorsing each other. To perform this task effectively we need to construct the neighbourhood graph of the result set. For a given subset of documents a neighbourhood graph consists of all links between the documents. It consists of the documents that appear in the retrieved document set.

For this study let us consider a web graph G(V,E) having the vertex set V and edge set E $\subseteq$ V X V .The result set of the query URLs is known as root set R $\subseteq$ V. From this the algorithms construct a neighbourhood graph which consists of base set B $\subseteq$ V (the root set and some of neighbouring vertices) and some of the edges in E are included in B. To define the neighbourhood graph formally sampling operator and link selection predicate are used.

For the given set A, the sampling operator $S_n[A]$ selects n elements uniformly at random from the set A ;

$$S_n[A] = A \; if \;\; | A | \leq n$$

For the given set a link selection predicate P uses an edge (u,v) $\in$ E. In this study three selection predicates are used.

$$all(u, v) \Leftrightarrow true$$

$$inter - host(u, v) \Leftrightarrow host(a) \neq host(b)$$

$$inter - domain(u, v) \Leftrightarrow domain(a) \neq domain(b)$$

where host(u) represents the host of URL a and domain(u) represents the domain of URL u. Therefore *all* is true for all the links in the results set whereas *inter-host* is true for inter-host links and *inter-domain* is true for inter-domain links.

The outlink set O of the root set with respect to link selection predicate P is represented as:

$$\boldsymbol{O^P} = \bigcup_{\boldsymbol{u \in R}} \{\, \boldsymbol{v} \in \boldsymbol{V} : (\boldsymbol{u}, \boldsymbol{v}) \in \boldsymbol{E} \wedge \boldsymbol{P}(\boldsymbol{u}, \boldsymbol{v})\}$$

The outlink set O of the root set with respect to link selection predicate and sampling operator S is represented as:

$$\boldsymbol{O_s^P} = \bigcup_{\boldsymbol{u \in R}} \boldsymbol{S_S}[\{\boldsymbol{v} \in \boldsymbol{V} : (\boldsymbol{u}, \boldsymbol{v}) \in \boldsymbol{E} \wedge \boldsymbol{P}\,(\boldsymbol{u}, \boldsymbol{v})\}]$$

The inlink set I of the root set with respect to link selection predicate P is represented as:

$$\boldsymbol{I^P} = \bigcup_{\boldsymbol{v \in R}} \{\boldsymbol{u} \in \boldsymbol{V} : (\boldsymbol{u}, \boldsymbol{v}) \in \boldsymbol{E} \wedge \boldsymbol{P}\,(\boldsymbol{u}, \boldsymbol{v})\}$$

The inlink set I of the root set with respect to link selection predicate and sampling operator S is represented as:

$$\boldsymbol{I_s^P} = \bigcup_{\boldsymbol{v \in R}} \boldsymbol{S_s}[\{\boldsymbol{u} \in \boldsymbol{V} : (\boldsymbol{u}, \boldsymbol{v}) \in \boldsymbol{E} \wedge \boldsymbol{P}\,(\boldsymbol{u}, \boldsymbol{v})\}]$$

The base set $\boldsymbol{B_s^P}$ of the root set R with respect to P and s is defined as:

$$\boldsymbol{B_s^P} = \boldsymbol{R} \cup \boldsymbol{I_s^P} \cup \boldsymbol{O^P}$$

The neighbourhood graph ($\boldsymbol{B_s^P}$, $\boldsymbol{N_s^P}$) has the base set $\boldsymbol{B_s^P}$ as its vertex set and an edge set $\boldsymbol{N_s^P}$ containing those edges in E that are covered by $\boldsymbol{B_s^P}$ and permitted by P:

$$\boldsymbol{N_s^P} = \{(\boldsymbol{u}, \boldsymbol{v}) \in \boldsymbol{E} : \boldsymbol{u} \in \boldsymbol{B_s^P} \wedge \boldsymbol{v} \in \boldsymbol{B_s^P} \wedge \boldsymbol{P}(\boldsymbol{u}, \boldsymbol{v})\}$$

4.2.2 Mean Average Precision

It is one of the most frequently used measures of a ranked retrieval run. To define MAP one needs to define Precision at position k (P@k). The Average precision of a given query is the arithmetic mean of the precision scores after each relevant document is retrieved. It takes into account both recall and precision oriented aspects. The precision P@k at document cut off value k is defined the fraction of relevant results among the k highest ranking results and is represented as $\frac{1}{k}\sum_{i=1}^{k} rel(i)$

The average precision at document cut off value of k[18] is defined to be :

$$\text{Average Precision @k} = \frac{\sum_{i=1}^{k} rel(i) P@i}{\sum_{i=1}^{n} rel(i)}$$

The mean average precision MAP@k at document cut off value k of a query set is the arithmetic mean of the average precisions of all queries in the query set.

Table 4. 1 MAP values for PageRank Algorithm

Vertex	Rank value	MAP value	Rank value	MAP Rank
	Normal Probability		**Conditional Probability**	
A	0.057379	0.004440618	0.226708	0.051396517
B	0.052711	0.004079357	0.085777	0.019446332
C	0.04867	0.00376662	0.079962	0.018128025
D	0.033931	0.002625954	0.012979	0.002942443
E	0.052657	0.004075178	0.086827	0.019684376
F	0.036669	0.002837851	0.02315	0.00524829
G	0.046346	0.003586763	0.075543	0.017126202
H	0.051025	0.003948876	0.019294	0.004374104
I	0.036291	0.002808597	0.017056	0.003866732
J	0.056604	0.00438064	0.028023	0.006353038
K	0.074617	0.005774684	0.04022	0.009118196
L	0.06314	0.004886468	0.131088	0.029718698
M	0.024737	0.001914421	0.010093	0.002288164
N	0.048196	0.003729937	0.028566	0.006476141
O	0.066418	0.005140155	0.010093	0.002288164
P	0.054547	0.004221447	0.041015	0.009298429
Q	0.077391	0.005989367	0.032421	0.0073501
R	0.038425	0.002973749	0.018262	0.004140141
S	0.029511	0.002283886	0.010093	0.002288164
T	0.050736	0.00392651	0.022829	0.005175517
SUM	1.000001	0.077391077	0.04999995	0.011335389
MAP		0.077391		0.226708

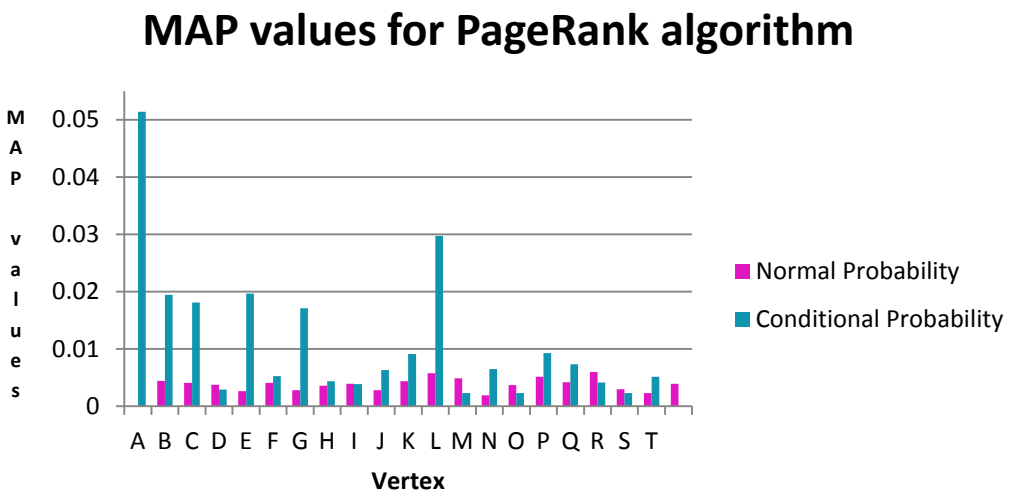


Figure 4.1 Mean Average Precision for the PageRank algorithm

The performance of the MAP values of the PageRank algorithm is far better than the values obtained with normal probability. We can use the conditional probability approach to obtain better results.

Table 4.2 MAP values for HITS Algorithm

Vertex	Rank value	MAP Rank	Rank value	MAP Rank
	Normal Probability		Conditional Probability	
A	0.184901	0.08288039	0.19985	0.08849018
B	0.152608	0.06840532	0.066065	0.02925246
C	0.090388	0.0405157	0.081123	0.03591989
D	0.368814	0.16531792	0.064201	0.02842711
E	0.301779	0.13527002	0.322223	0.14267487
F	0.095192	0.04266905	0.442783	0.19605679
G	0.122821	0.05505353	0.112027	0.04960365
H	0.448242	0.20092089	0.017745	0.00785718
I	0.08625	0.03866087	0.071056	0.03146239
J	0.204792	0.09179638	0.102881	0.04555396
K	0.128007	0.05737811	0.04859	0.02151483

L	0.326829	0.14649848	0.382678	0.16944331
M	0.053757	0.02409615	0.35755	0.15831706
N	0.167077	0.07489093	0.107267	0.047496
O	0.074019	0.03317842	0.248195	0.10989653
P	0.312388	0.14002542	0.373748	0.16548926
Q	0.167396	0.07503392	0.190846	0.08450336
R	0.267497	0.11990339	0.020567	0.00910672
S	0.045225	0.02027174	0.00586	0.00259471
T	0.283849	0.12723304	0.300431	0.13302574
SUM	3.881831	1.73999969	3.515686	1.55668599
MAP		0.448242		0.442783

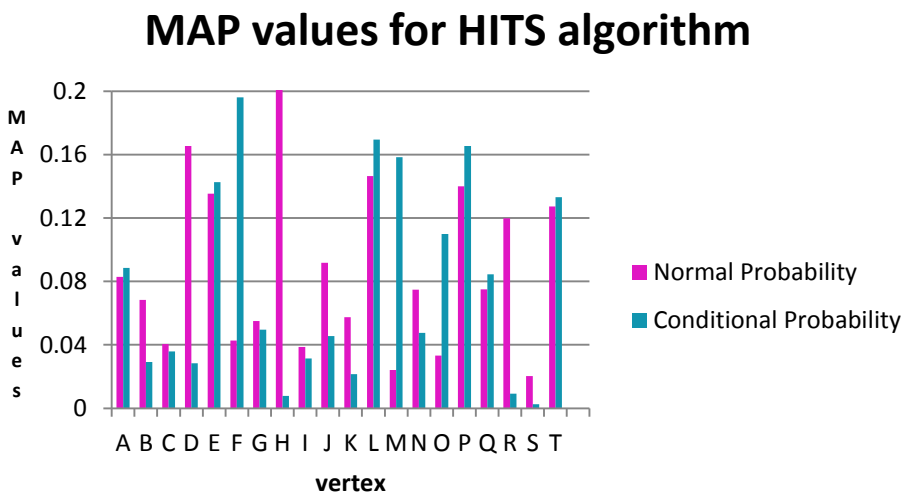


Figure 4.2 Mean Average Precision for the HITS algorithm

The MAP values for normal and conditional probability of HITS algorithms are compared. The values obtained in the case of normal probability are almost similar to the values obtained in the case of conditional probability.

Table 4.3 MAP values for SALSA Algorithm

Vertex	Rank Value	MAP Rank	Rank Value	MAP Rank
	Normal Probability		**Conditional Probability**	
A	0.340917	0.12741671	0.19416	0.14595027
B	0.260443	0.09733979	0.000734	0.00055175
C	0.168371	0.06292816	0.002724	0.00204763
D	0.305857	0.11431314	0.001251	0.00094038
E	0.192643	0.07199974	0.00008	6.0136E-05
F	0.228572	0.0854281	0.001012	0.00076072
G	0.118198	0.04417615	0.00009	6.7653E-05
H	0.178553	0.06673365	0.004195	0.00315339
I	0.373747	0.13968682	0.007417	0.00557537
J	0.123609	0.04619849	0.001482	0.00111402
K	0.249881	0.09339227	0.004659	0.00350217
L	0.304532	0.11381792	0.186012	0.13982541
M	0.227	0.08484057	0.000057	4.2847E-05
N	0.148242	0.055405	0.000584	0.00043899
O	0.179108	0.06694108	0.000452	0.00033977
P	0.297933	0.11135156	0.751701	0.56505439
Q	0.066701	0.0249293	0.007211	0.00542052
R	0.128083	0.04787064	0.0051	0.00383368
S	0.117946	0.04408196	0.568286	0.42718115
T	0.139901	0.05228758	0.198779	0.14942237
SUM	4.150237	1.55113863	1.935986	1.45528261
MAP		0.373747		0.751701

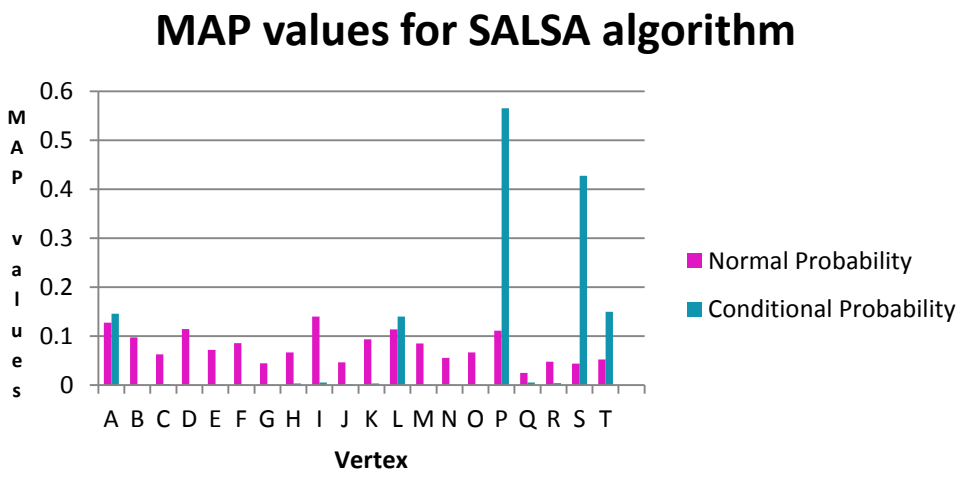


Figure 4.3 Mean Average Precision for the SALSA algorithm

The MAP values for SALSA algorithm are compared. Though the higher values corresponding to conditional probability are smaller in number but when these are taken as a whole these are marginally higher than the values obtained in the case of normal probability

Table 4.4 MAP values for NORM(P) Algorithm

Vertex	Rank Value	MAP Rank	Rank Value	MAP Rank
	Normal Probability		Conditional Probability	
A	0.147144	0.04933076	0.231371	0.11187019
B	0.141018	0.04727699	0.023093	0.0111657
C	0.090682	0.03040159	0.085937	0.0415514
D	0.294528	0.09874198	0.191776	0.09272561
E	0.209551	0.07025302	0.295197	0.1427307
F	0.146371	0.04907161	0.113298	0.05478072
G	0.267836	0.08979336	0.106789	0.05163355
H	0.209468	0.07022519	0.003266	0.00157914
I	0.168579	0.05651695	0.249036	0.1204114
J	0.194781	0.0653013	0.192517	0.09308389
K	0.220413	0.07389456	0.229849	0.11113429

L	0.235336	0.07889757	0.482962	0.23351696
M	0.156053	0.05231755	0.002309	0.00111642
N	0.335255	0.11239592	0.039153	0.01893087
O	0.294528	0.09874198	0.11686	0.05650298
P	0.255165	0.08554534	0.348624	0.16856319
Q	0.2978	0.09983894	0.48351	0.23378192
R	0.236898	0.07942124	0.161403	0.07803996
S	0.213637	0.07162287	0.082413	0.03984751
T	0.249007	0.08348084	0.006174	0.00298519
SUM	4.36405	1.46306958	3.445537	1.66595159
MAP		0.335255		0.48351

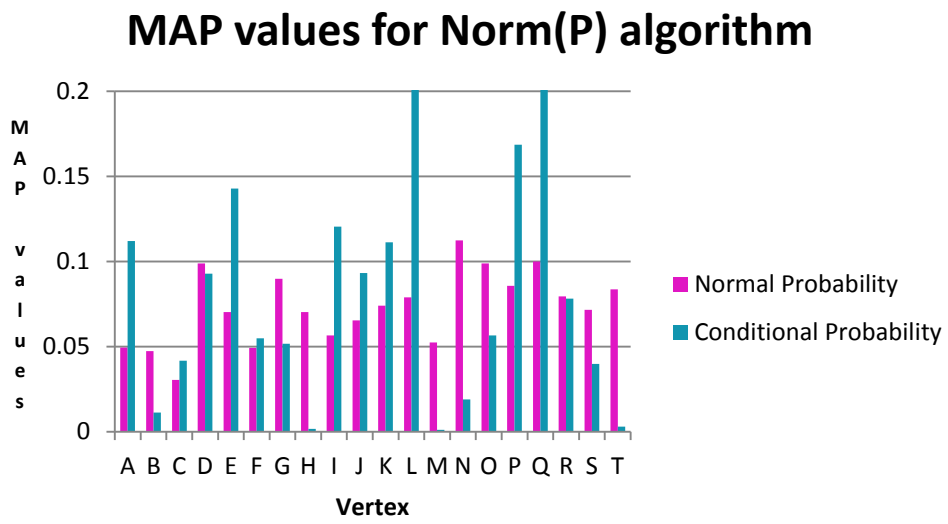


Figure 4.4 Mean Average Precision for Norm(P) algorithm

The results obtained in the case of MAP value for conditional probability are more efficient than the corresponding values of the normal probability in the case of Norm (P) algorithm.

4.2.3 Mean Reciprocal Rank

The Reciprocal Rank (RR) of the result set of query is defined as the reciprocal value of the highest ranking relevant document's rank in the result set[19]. For a query q, the rank positions of its relevant retrieved document is presented as r_1.Thus the Mean Reciprocal rank is represented as $1/r_1$

$$RR@k = \begin{cases} \frac{1}{i} & if\ \exists i\ \leq k : rel(i) = 1\ \wedge\ \forall j\ < i : rel(j) = 0 \\ 0 & otherwise \end{cases}$$

We can define the Mean Reciprocal Rank of a query set is the average reciprocal rank of all queries in the query sets[8]. Let there are n number of queries in the query set. It is represented as

$$MRR = \frac{1}{n} \sum_{k=1}^{n} RR@k$$

Table 4.5 MRR values for PageRank Algorithm

Vertex	Rank Value	Reciprocal rank	Rank Value	Reciprocal Rank
	Normal Probability		Conditional Probability	
A	0.057379	0.741416961	0.226708	1
B	0.052711	0.681099869	0.085777	0.378358946
C	0.04867	0.628884496	0.079962	0.352709212
D	0.033931	0.438435994	0.012979	0.057249854
E	0.052657	0.680402114	0.086827	0.382990455
F	0.036669	0.473814785	0.02315	0.102113732
G	0.046346	0.598855164	0.075543	0.333217178
H	0.051025	0.659314391	0.019294	0.085105069
I	0.036291	0.468930496	0.017056	0.07523334
J	0.056604	0.731402876	0.028023	0.123608342
K	0.074617	0.964156039	0.04022	0.177408825
L	0.06314	0.815857141	0.131088	0.578223971
M	0.024737	0.31963665	0.010093	0.044519823
N	0.048196	0.622759752	0.028566	0.126003493

O	0.066418	0.858213487	0.010093	0.044519823
P	0.054547	0.704823558	0.041015	0.180915539
Q	0.077391	1	0.032421	0.143007746
R	0.038425	0.496504762	0.018262	0.080552958
S	0.029511	0.38132341	0.010093	0.044519823
T	0.050736	0.655580106	0.022829	0.100697814
	SUM	12.92141205	**SUM**	4.410955943
	MRR	0.646070603	**MRR**	0.220547797

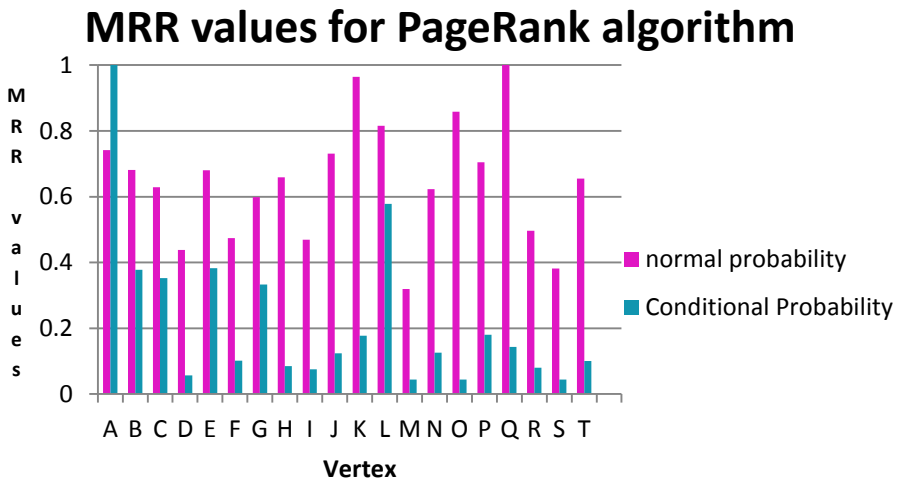


Figure 4.5 Mean Reciprocal Rank for the PageRank algorithm

In most of the cases normal probability perform better in the MRR values as compared to the conditional probability.

Table 4.6 MRR values for HITS Algorithm

Vertex	**Rank Value**	**Reciprocal rank**	**Rank Value**	**Reciprocal Rank**
	Normal Probability		**Conditional Probability**	
A	0.184901	0.412502621	0.19985	0.451349758
B	0.152608	0.340458949	0.066065	0.149204012

C	0.090388	0.201650002	0.081123	0.183211641
D	0.368814	0.822801076	0.064201	0.144994275
E	0.301779	0.673250164	0.322223	0.727722157
F	0.095192	0.212367427	0.442783	1
G	0.122821	0.274006006	0.112027	0.253006552
H	0.448242	1	0.017745	0.040076064
I	0.08625	0.192418381	0.071056	0.160475899
J	0.204792	0.456878204	0.102881	0.232350836
K	0.128007	0.285575649	0.04859	0.109737727
L	0.326829	0.729135155	0.382678	0.864256306
M	0.053757	0.119928521	0.35755	0.80750616
N	0.167077	0.372738387	0.107267	0.242256365
O	0.074019	0.165131781	0.248195	0.560534167
P	0.312388	0.696918183	0.373748	0.844088414
Q	0.167396	0.373450056	0.190846	0.431014741
R	0.267497	0.596769156	0.020567	0.046449389
S	0.045225	0.10089416	0.00586	0.013234474
T	0.283849	0.63324945	0.300431	0.678506176
	SUM	8.660123326	**SUM**	7.939975112
	MRR	0.433006166	**MRR**	0.396998756

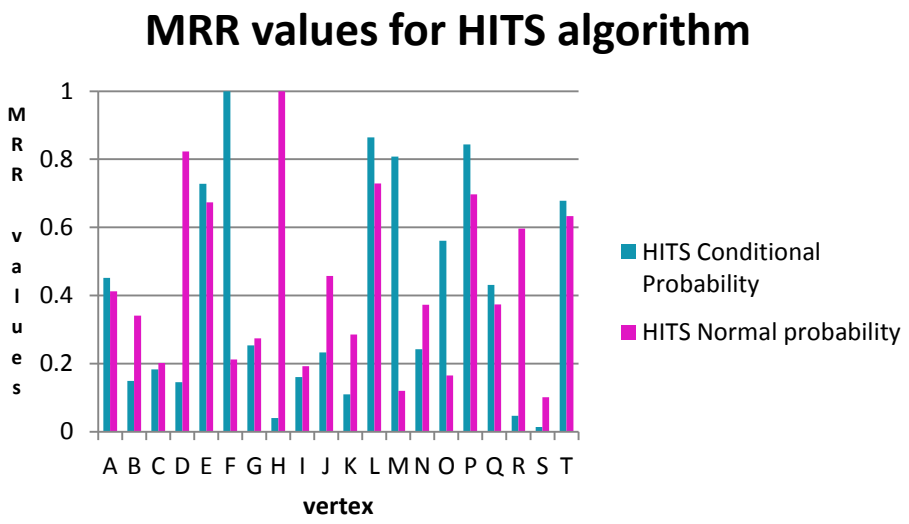


Figure 4.6 Mean Reciprocal Rank for the HITS algorithm

The MRR values of HITS algorithms the performance of conditional probability values is almost similar to normal probability values. Here the values are conditional probability when taken as whole is slightly less than the normal values taken as whole.

Table 4.7 MRR values for SALSA Algorithm

Vertex	Rank Value	Reciprocal rank	Rank Value	Reciprocal Rank
	Normal Probability		Conditional Probability	
A	0.340917	0.91215983	0.19416	0.258294189
B	0.260443	0.696843052	0.000734	0.000976452
C	0.168371	0.450494586	0.002724	0.003623781
D	0.305857	0.818353057	0.001251	0.001664226
E	0.192643	0.515436913	0.00008	0.000106425
F	0.228572	0.611568789	0.001012	0.00134628
G	0.118198	0.316251368	0.00009	0.000119728
H	0.178553	0.477737614	0.004195	0.005580676
I	0.373747	1	0.007417	0.009866955
J	0.123609	0.330729076	0.001482	0.001971529
K	0.249881	0.668583293	0.004659	0.006197943

L	0.304532	0.814807878	0.186012	0.247454773
M	0.227	0.607362735	0.000057	7.5828E-05
N	0.148242	0.396637297	0.000584	0.000776905
O	0.179108	0.479222576	0.000452	0.000601303
P	0.297933	0.797151549	0.751701	1
Q	0.066701	0.178465647	0.007211	0.00959291
R	0.128083	0.342699741	0.0051	0.006784612
S	0.117946	0.315577115	0.568286	0.756000059
T	0.139901	0.374320061	0.198779	0.264438919
	SUM	11.10440218	**SUM**	2.575473493
	MRR	0.555220109	**MRR**	0.128773675

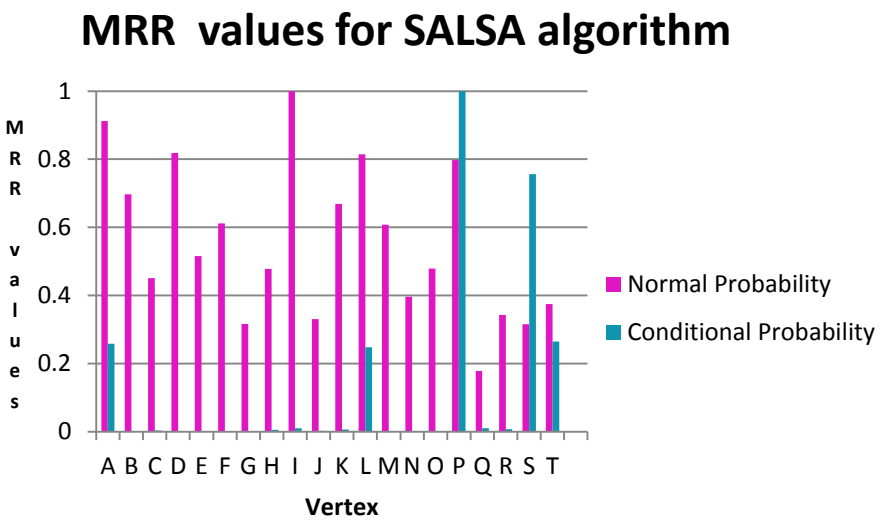


Figure 4.7 Mean Reciprocal Rank for the SALSA algorithm

MRR values of the normal probability clearly dominate the MRR values of the conditional probability.

Table 4.8 MRR values for NORM(P) Algorithm

Vertex	Rank Value	Reciprocal rank	Rank Value	Reciprocal Rank
	Normal Probability		**Conditional Probability**	
A	0.147144	0.438901732	0.231371	0.478523712
B	0.141018	0.420629073	0.023093	0.047761163
C	0.090682	0.270486644	0.085937	0.177735724
D	0.294528	0.878519336	0.191776	0.396632955
E	0.209551	0.625049589	0.295197	0.610529255
F	0.146371	0.436596024	0.113298	0.234324006
G	0.267836	0.798902328	0.106789	0.22086203
H	0.209468	0.624802016	0.003266	0.006754772
I	0.168579	0.502838138	0.249036	0.515058634
J	0.194781	0.580993572	0.192517	0.398165498
K	0.220413	0.657448808	0.229849	0.475375897
L	0.235336	0.701961194	0.482962	0.998866621
M	0.156053	0.465475534	0.002309	0.004775496
N	0.335255	1	0.039153	0.080976609
O	0.294528	0.878519336	0.11686	0.241690968
P	0.255165	0.761107217	0.348624	0.721027487
Q	0.2978	0.888279071	0.48351	1
R	0.236898	0.706620334	0.161403	0.333815226
S	0.213637	0.637237327	0.082413	0.170447354
T	0.249007	0.742739109	0.006174	0.012769126
	SUM	13.01710638	**SUM**	7.126092532
	MRR	0.650855319	**MRR**	0.356304627

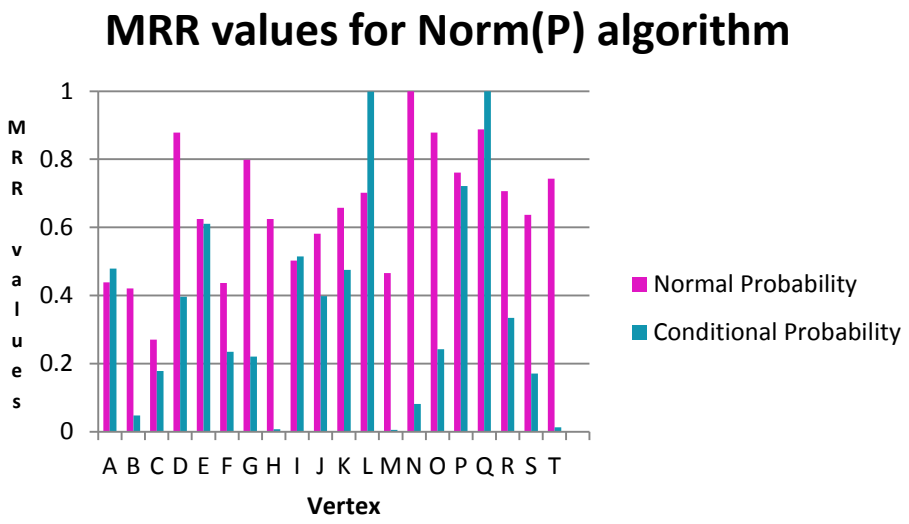


Figure 4.8 Mean Reciprocal Rank for the Norm(P) algorithm

The MRR values of Norm (P) algorithm the results of the normal probability are better as compared to conditional probability.

4.2.4. Discounted Cumulated Gain

For the calculation of Discounted Cumulated Gain we first need to find the cumulated gain. In this relevance score of the given document is calculated as a gained value measure for its ranked position in the result set and the calculated gain is computed by summing from the position 1 to the last. Now the ranked document IDs are replaced by their relevance score thus ranked document lists become the gained value lists. Assuming the relevance scores 0 to 3 are used (3 denoting the highest value, 0 being the lowest). The average of the highest ranked value and the lowest ranked value is taken say it is σ_1.The average of σ_1 and the lowest ranked value is taken say it is σ_2. The average of σ_1 and the highest ranked value is taken say it is σ_3.The range of ranked values between σ_2 and the lowest ranked value has the relevance as 0. The range of ranked values between σ_2 and σ_1 has the relevance as 1. The range of ranked values between σ_1 and σ_3 has the relevance as 2. The range of ranked values between σ_3 and the highest ranked value has the relevance as 3.Let G[i] denotes the gain vector G at a position *i*. Cumulated Gain vector CG is defined recursively as[20]

$$CG[i] = \begin{cases} G[1], & if\ i = 1 \\ CG[i-1] + G[i], & otherwise \end{cases}$$

In the case of Discounted Cumulated Gain a discounting function is required that continuously and steadily reduces the document score when its rank increases. This is required to allow user persistence in evaluating the further documents. The most common way to perform this task is to divide the document score by the log of its rank. If b denotes the base of the logarithm, The Discounted Cumulated Gain is defined as follows[20]:

$$DCG[i] = \begin{cases} CG[i], & if\ i < b \\ DCG[i-1] + \dfrac{G[i]}{b_{\log i}}, & if\ i \geq b \end{cases}$$

4.2.5 Normalised Discounted Cumulative Gain

The normalised discounted Cumulated Gain nDCG@k of a query set is obtained by dividing the DCG@k of the result set rank ordered according to the scores by the DCG@k of the result set rank ordered according to the ideal scoring function. The ideal scoring function is the one that rank order results according to their ratings. This ideal scoring function is obtained by sorting documents of the result list according to the relevance thereby giving the maximum possible value of DCG till the position k and this DCG is known as Ideal Discounted Cumulative Gain IDCG@k[20]. For the given query the Normalised Discounted Cumulative Gain is given by:

$$nDCG@k = \frac{DCG@k}{IDCG@k}$$

Table 4.9 NDCG values for PageRank Algorithm

Vertex	Rank Value	NDCG values	Rank Value	NDCG values
	Normal Probability		Conditional Probability	
A	0.057379	0.666667	0.226708	1
B	0.052711	0.5	0.085777	0.8
C	0.04867	0.460031	0.079962	0.822409
D	0.033931	0.4083	0.012979	0.755339
E	0.052657	0.416398	0.086827	0.771397
F	0.036669	0.385796	0.02315	0.72845
G	0.046346	0.393035	0.075543	0.779714
H	0.051025	0.410516	0.019294	0.779714
I	0.036291	0.399623	0.017056	0.779714
J	0.056604	0.439144	0.028023	0.779714
K	0.074617	0.498463	0.04022	0.779714
L	0.06314	0.531293	0.131088	0.860004
M	0.024737	0.520273	0.010093	0.860004
N	0.048196	0.529753	0.028566	0.860004
O	0.066418	0.587527	0.010093	0.860004
P	0.054547	0.625146	0.041015	0.860004
Q	0.077391	0.680368	0.032421	0.860004
R	0.038425	0.680368	0.018262	0.860004
S	0.029511	0.680368	0.010093	0.860004
T	0.050736	0.697777	0.022829	0.860004

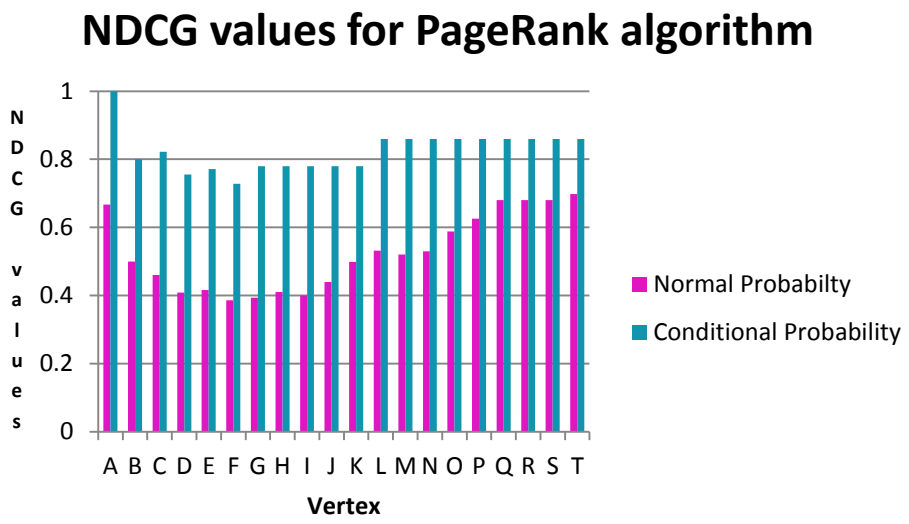


Figure 4.9 Normalised Discounted Cumulative Gain for the PageRank algorithm

The NDCG values of the PageRank algorithm are compared. In this case the NDCG values for the conditional probability are relatively higher than their corresponding values.

Table 4.10 NDCG values for HITS Algorithm

Vertex	Rank Value	NDCG values	Rank Value	NDCG values
	Normal Probability		Conditional Probability	
A	0.184901	0.333333	0.19985	0.333333
B	0.152608	0.333333	0.066065	0.166667
C	0.090388	0.275412	0.081123	0.126698
D	0.368814	0.423633	0.064201	0.106465
E	0.301779	0.47805	0.322223	0.181522
F	0.095192	0.440678	0.442783	0.274025
G	0.122821	0.411086	0.112027	0.257397
H	0.448242	0.489949	0.017745	0.250291
I	0.08625	0.476221	0.071056	0.243918
J	0.204792	0.489861	0.102881	0.243918
K	0.128007	0.47791	0.04859	0.243918

L	0.326829	0.51292	0.382678	0.311464
M	0.053757	0.51292	0.35755	0.376901
N	0.167077	0.534578	0.107267	0.376901
O	0.074019	0.534578	0.248195	0.418221
P	0.312388	0.575808	0.373748	0.478759
Q	0.167396	0.595982	0.190846	0.498506
R	0.267497	0.635531	0.020567	0.498506
S	0.045225	0.635531	0.00586	0.498506
T	0.283849	0.67369	0.300431	0.535858

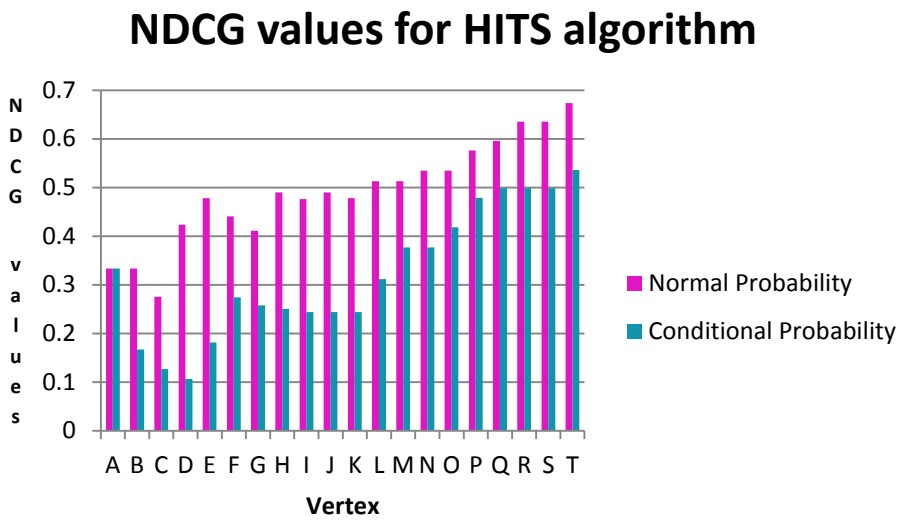


Figure 4.10 Normalised Discounted Cumulative Gain for the HITS algorithm

The performance for the NDCG values in the case of normal probability values is better than the values of the conditional probability in the HITS algorithm.

Table 4. 11 NDCG values for SALSA Algorithm

Vertex	Rank Value	NDCG values	Rank Value	NDCG values
	Normal Probability		Conditional Probability	
A	0.340917	1	0.19416	0.333333
B	0.260443	0.833333	0.000734	0.166667

C	0.168371	0.713427	0.002724	0.137706
D	0.305857	0.759192	0.001251	0.128835
E	0.192643	0.707696	0.00008	0.128835
F	0.228572	0.727433	0.001012	0.128835
G	0.118198	0.684854	0.00009	0.128835
H	0.178553	0.675254	0.004195	0.128835
I	0.373747	0.713889	0.007417	0.128835
J	0.123609	0.698282	0.001482	0.128835
K	0.249881	0.725047	0.004659	0.128835
L	0.304532	0.769307	0.186012	0.128835
M	0.227	0.792074	0.000057	0.128835
N	0.148242	0.795747	0.000584	0.128835
O	0.179108	0.812959	0.000452	0.128835
P	0.297933	0.863395	0.751701	0.225461
Q	0.066701	0.863395	0.007211	0.225461
R	0.128083	0.863395	0.0051	0.225461
S	0.117946	0.863395	0.568286	0.316448
T	0.139901	0.863395	0.198779	0.376068

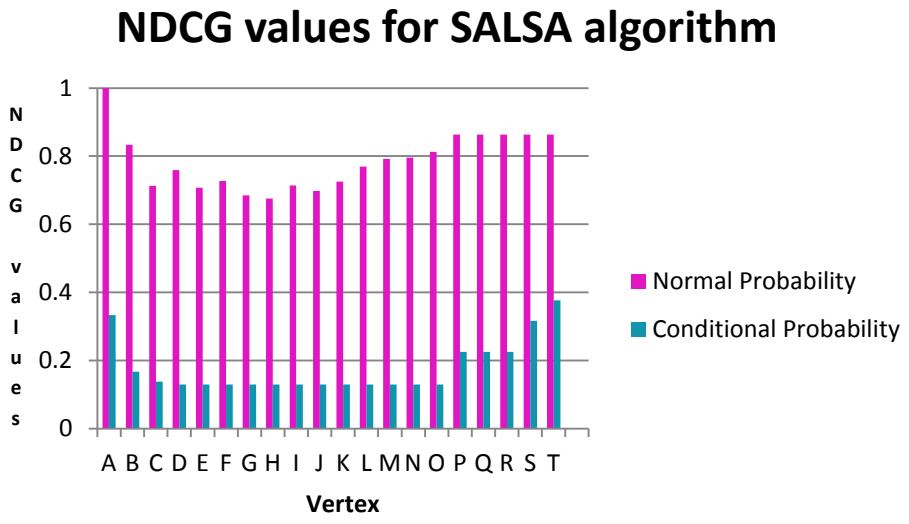


Figure 4.11 Normalised Discounted Cumulative Gain for the SALSA algorithm

The NDCG values for SALSA algorithm of the normal probability are comparatively higher than their corresponding values in the case of conditional probability.

Table 4. 12 NDCG values for NORM(P) Algorithm

Vertex	Rank Value	NDCG values	Rank Value	NDCG values
	Normal Probability		Conditional Probability	
A	0.147144	0	0.231371	0.333333
B	0.141018	0	0.023093	0.166667
C	0.090682	0	0.085937	0.137706
D	0.294528	0.097448	0.191776	0.181557
E	0.209551	0.16734	0.295197	0.258829
F	0.146371	0.156457	0.113298	0.2483
G	0.267836	0.204116	0.106789	0.239336
H	0.209468	0.244082	0.003266	0.231514
I	0.168579	0.233009	0.249036	0.284571
J	0.194781	0.22334	0.192517	0.304482
K	0.220413	0.253101	0.229849	0.331208
L	0.235336	0.279735	0.482962	0.408577
M	0.156053	0.270394	0.002309	0.408577
N	0.335255	0.367758	0.039153	0.408577
O	0.294528	0.399387	0.11686	0.408577
P	0.255165	0.430278	0.348624	0.454804
Q	0.2978	0.460509	0.48351	0.522661
R	0.236898	0.490142	0.161403	0.544833
S	0.213637	0.519231	0.082413	0.544833
T	0.249007	0.547822	0.006174	0.544833

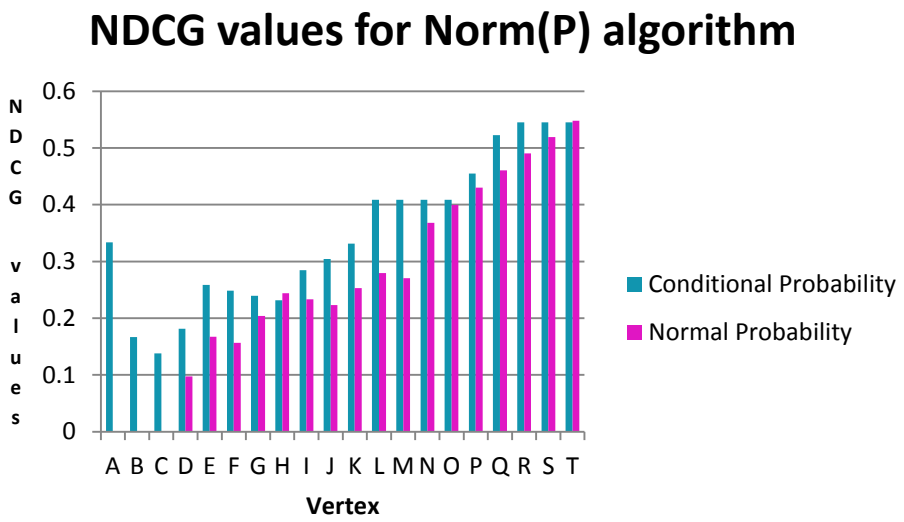


Figure 4.12 Normalised Discounted Cumulative Gain for the PageRank algorithm

NDCG values of the conditional probability clearly dominate the NDCG values of the normal probability in the Norm (P) algorithm.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

With this study we propose that instead of considering only the concerned outlink on a given page we should consider all the outlinks .Based on these outlinks we should calculate the probability of visiting the given outlink and this probability is used in the adjacency matrix of the web graph. In the case of MRR values the performance these probability values calculations are not very sound as compared to normal scenario of considering the outlink only. The performance of MAP values is considered to be same in both the scenario. For the NDCG values PageRank algorithm and Norm (P) algorithm perform exceptionally well in case of conditional probability. It is also explained how the relevance of a document is related to ranking.

The results obtained so can be further optimised by using various probability functions like those of probability mass function, probability density function.

REFERENCES

[1] A.Gulli, A.Signorini," The Indexable web is More than 11.5 Billion pages", Proceedings of the 14thWorld Wide Web Conference, Pages 902 - 903 , 2005

[2] A.Broder, "Web Searching Technology Overview in Advanced school and Workshop on Models and Algorithms for the World Wide Web",Udine,Italy , 2002

[3] B.J.Jansen, A.Spink, J.Bateman, T.Saracevic, "Real life Information Retrieval: A study of user queries on the web", ACM SIGIR Forum, 1998

[4] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, S.Stata, A. Tomkins, and J. Wiener. "Graph structure in the web.Computer Networks", volume 33 pages 309–320, 2000.

[5] Monika Henzinger,"Link Analysis in Web Information Retrieval", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2000

[6]M. Marchiori. "The quest for correct information on Web: Hyper search engines" ,Proceedings of the 6th International World Wide Web Conference, 1997

[7] Mei Kobayashi and Koichi Takeda," Information Retrieval on the Web", ACM Computing Surveys, Vol. 32, No. 2, June 2000

[8]Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd, " The PageRank citation ranking: Bringing order to the web." Technical report, Stanford Digital Library Technologies Project, Stanford University, Stanford, CA, USA, 1998.

[9] Allan Borodin, Gareth O. Roberts, Jeffrey S. Rosenthal and Panayiotis Tsaparas , "Link Analysis Ranking: Algorithms, Theory and Experiments ", ACM Transactions on

Internet Technology (TOIT) Volume 5 Issue 1, February 2005 Pages 231 - 297 February 2005

[10] J. Kleinberg "Hubs , Authorities and Communities" ACM Computing Surveys (CSUR), volume 31,Issue 4,December 1999

[11] J. Kleinberg, "Authoritative sources in a hyperlinked environment", Journal of ACM (JASM), Volume 46 Issue 5, Sept. 1999, Pages 604 - 632 1999

[12]Amy N. Langville, Carl D. Meyer, "A Survey of Eigenvector Methods of Web Information Retrieval",Society For Industrial And Applied Mathematics, volume 47,Issue 1 ,March 2005,page 135-161

[13]Jianjun Yang, Zongming Fei, "A new channel assignment algorithm for wireless mesh networks", International Journal of Pervasive Computing and Communications, Vol. 5, Issue 3, pages 233 – 248,(2009).

[14]R. Lempel and S. Moran , "SALSA: The Stochastic Approach for Link- Structure Analysis", ACM Transactions on Information Systems(TOIS),volume 19 Issue 2,April 2001,pages 131-160

[15]Ayman Farahatz, Thomas Lofaro, Joel C. Millerk, Gregory Rae And Lesley A. Ward, "Authority Rankings from HITS, PageRank, and SALSA", Society for Industrial and Applied Mathematics,Vol. 27, Issue 4, Pages 1181–1201

[16] Cynthia Rudin," Ranking with a P-Norm Push" ,Springer-Verlag Berlin Heidelberg 2006

[17] Richard A. Johnson, Irwin Miller and John Freund ,"Probability and Statistics for Engineers", New Jersey : Pearson Education,2011

[18] Tie-Yan Liu, "Learning to Rank for Information Retrieval", ACM SIGIR 2008

[19] E.M. Voorhees , “TREC-8 Question Answering Track Report,Proceedings of the 8th Text Retrieval Conference National Institute of Standards and Technology (NIST)”. 1999

[20] Kalervo Jarvelin and Jaana Kekalaine ,Cumulated Gain-Based Evaluation of IR Techniques, ACM Transactions on Information Systems, Vol. 20, Issue 4, October 2002, Pages 422–446

APPENDICES

Webgraph with Present scenario of considering the outlinks only

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	0	1	1	0	1	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0
B	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
C	0	0	0	1	0	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0
D	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	1	0
E	1	0	0	1	0	0	1	0	0	1	0	0	1	1	0	0	0	0	1	0
F	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
G	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0
H	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0
I	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0
J	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1
K	0	0	1	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0
L	1	0	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
N	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0
O	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
P	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0	0
R	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	1
S	0	1	0	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0
T	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0

Webgraph with the proposed solution

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	**0**	**0.166**	**0.166**	**0**	**0.166**	**0**	**0.1666**	**0**	**0**	**0**	**0**	**0.166**	**0**	**0**	**0**	**0.1666**	**0**	**0**	**0**	**0**
B	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0.055**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0.055**
C	**0**	**0**	**0**	**0.0208**	**0**	**0.0208**	**0**	**0.0208**	**0.0208**	**0**	**0.02008**	**1**	**0**	**0.0208**	**0**	**0**	**0**	**0.0208**	**0**	**0**
D	**0**	**1**	**0**	**0**	**0**	**0**	**0.00416**	**0**	**0**	**0**	**0.00416**	**0**	**0**	**0**	**0**	**0.00416**	**0**	**0**	**0.00416**	**0**
E	**1**	**0**	**0**	**0.0238**	**0**	**0**	**1**	**0**	**0**	**0.0238**	**0**	**0**	**0.0238**	**0.0238**	**0**	**0**	**0**	**0**	**0.0238**	**0**
F	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
G	**0**	**0**	**0**	**0.0416**	**0**	**0**	**0**	**0**	**0**	**0**	**0.04016**	**0**	**0**	**0**	**0**	**0**	**0**	**0.0416**	**0.04016**	**0**
H	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0.022**	**0**	**0**	**1**	**0.022**	**0**	**0**	**0**
I	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0.0052**	**0**	**0**	**0.0052**	**0**	**0**
J	**1**	**0**	**1**	**0**	**0**	**0.0114**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0.0114**
K	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**
L	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**0.033**	**0**	**0.033**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
M	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**
N	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0.0089**	**0**	**0**	**1**	**0**	**0**
O	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0.0337**	**0**	**0**	**0**

P	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0.055**	**0**	**0**	**0**	**0**	**0**	**0.055**	**0**	**0**	**0**	**0**	**0**
Q	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**0**
R	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0.169**	**0**	**0**	**0**	**0**	**1**
S	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0.133**	**0**	**0**	**0**	**0**	**0**
T	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0.364**	**0**	**1**	**0**	**0**	**0**

LIST OF PUBLICATIONS

Preet Kamal and Ravinder Kumar "Link-Based Web Ranking Algorithms based on weighted graph using probabilistic approach",Journal of Web Engineering