Comparative Analysis Of Pagerank And HITS Algorithms

Nidhi Grover
MCA Scholar
Institute of Information Technology and Management

Ritika Wason
Assistant professor, Dept. of Computer Sciences
Institute of Information Technology and Management

Abstract

The World Wide Web is expanding day by day and so is the amount of data available on web. In such a situation in order to trace relevant data, users mainly rely on varied search engines for finding suitable answers for their queries. This common trend has resulted in a rise in the number as well as use of different search engines. The present state of affairs necessitates comparison and analysis of different link analysis algorithms employed by search engines for ranking web pages against user queries. In this paper, we compare two popular web page ranking algorithms namely: HITS algorithm and PageRank algorithm. The paper highlights their variations, respective strengths, weaknesses and carefully analyzes both these algorithms using simulations developed for both

1. Introduction

In general, the World Wide Web (www) is a system of interlinked hypertext documents [1]. WWW provides an architectural framework for accessing linked documents spread out over millions of machines all over the Internet [3]. Retrieving useful information from the vast sea of data on World Wide Web has been one of the most challenging tasks. Web search engines have surfaced as a useful technique that helps in searching for useful information on the World Wide Web using search strings provided by the user. The search results of a search engine are generally presented as a list often referred to as search engine results pages (SERPs). To locate any information from the web, the user accesses his favorite search engine, issues queries and clicks on the returned pages [7]. The search results returned by a search engines are a mixture of large amount of relevant and irrelevant information [5]. Any user cannot read all web pages returned in response to the user's query. Hence, search engines help users trace relevant pages worth considering by displaying the resultant pages in a ranked order using different page rank algorithms [5]. Web-page ranking is a search-engine optimization technique used by search engines for ranking hundred thousands of web pages in a relative order of importance. Conventional search engine technology can be broadly classified into two main categories of search engines: the crawler based engine and the human-powered directories based engine [6]. A human-powered directory, for instance the Open Directory depends on humans for its listings [2]. The web pages in such a setting are stored in different directories on the basis of their category. When a query is fired, it is categorized first and then the appropriate directory is searched to locate the web page. They are constructed when the owner of a website submits the site for a review along with a short description of the site [6]. A search is generally based on the matches only in the descriptions submitted.

Crawler-based search engines, for instance Google, create their listings automatically [2]. They "crawl" or "spider" the web, to search for pages matching user requests. Once they generate result sets, people can navigate through the results. Crawler-based search engines retrieve contents of web pages using indexers [2]. Indexers are used to store and index information regarding retrieved pages. The Ranker determines the importance of web pages returned and the Retrieval Engine performs lookups on index tables.

The web page ranking algorithms play their role at the last component [16]. Exactly what information the user wants is unpredictable. So the web page ranking algorithms are designed to anticipate the user requirements from various static (e.g., number of hyperlinks, textual content) and dynamic (e.g., popularity) features [16]. They are important factors for making one search engine better than another [16]. Web search ranking algorithms play an important role in ranking web pages so that the user could get the good result which is more relevant to the user's query [8]. Figure 1 [4] illustrates the working of a typical search engine, which shows the flow graph for a searched query by a web user.

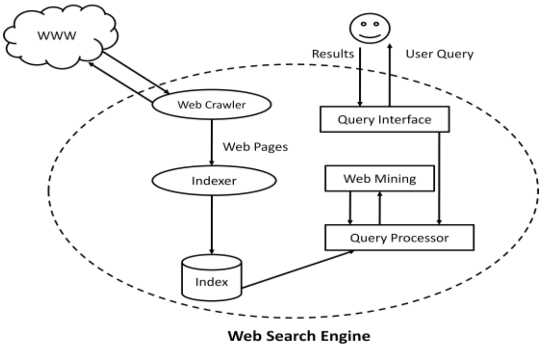


Fig 1: Working of Search Engine

The motive behind this paper to analyze the popular web page ranking algorithms- HITS algorithm and PageRank algorithm, their variations and to provide a comparative study of both and to highlight their relative strengths and limitations.

2. Literature Review

Hyperlink Analysis by itself is a part of a bigger research domain -Web Mining, which can be described as the process of applying data mining techniques to extract useful information from Web data [22]. Web mining helps the internet user about the web pages to be viewed in future [4]. The kinds of data that can be collected and used in Web Mining analysis include content data, structure data, and usage data [23]. The field of Web Mining can be broadly divided into three distinct categories, according to the kinds of data to be mined [23]:

1. **Web Content Mining (WCM):** WCM is responsible for exploring the proper and relevant information from the contents of web pages [4]. Content data corresponds to the collection of facts a Web page was designed to convey to the users [22]. It may consist of text, images, audio, video, or structured records such as lists and tables [22].

2. **Web Structure Mining (WSM):** The structure of a typical Web graph consists of Web pages as nodes, and hyperlinks as edges connecting two related pages[22]. WSM is used to find out the relation between different web pages by processing the structure of web [4]. Web Structure Mining is useful for extracting structure information from the Web. WSM can be performed at two levels:
    i. **Document structure analysis:** deals with the structure of a document such as the Document Object Model.
    ii. **Link type analysis:** deals with links that may be inter-document or intra-document.

The number of outlinks i.e. links from a page and the number of inlinks i.e. links to a page are very important parameters in the area of web mining [4]. As shown in the figure 2[22], the basis for hyperlink analysis is the inter-document link type structure. Hyperlinks provide structural information which, coupled with Web content, can be used to mine useful information from the Web and to measure the quality of information [22]. So WSM becomes a very important area to be researched in the field of web mining [4].

3. **Web Usage Mining (WUM):** Web Usage Mining is the application of data mining techniques to discover interesting usage patterns from Web data, in order to understand and better serve the needs of Web-based applications [23]. WUM is responsible for recording the user profile and user behavior inside the log file of the web [4]. Some of the typical usage data collected at a Web site include IP addresses, page references, and access time of the users [22].

The high level taxonomy of various research activities in Web Mining along with web structure mining is illustrated in figure2[22] below:

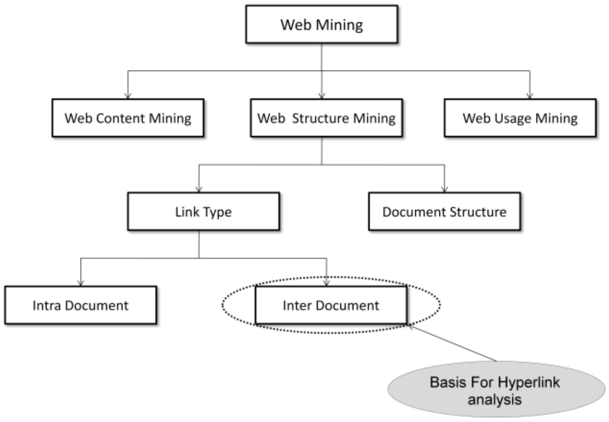


Fig 2: High level taxonomy of Web Mining

Many algorithms have been proposed in the area of web structure mining. They may be text-based or link based algorithms. The most popular class of algorithms have been link based algorithms namely, HITS algorithm developed by Jon Kleinberg in 1998 and PageRank algorithm originally developed at Stanford University by Larry Page and Sergey Brin in 1996. We discuss the structure and semantics of these algorithms in the following section.

3. Ranking Algorithms

The web page ranking algorithms rank the search results depending upon their relevance to the search query. For this algorithms rank the search results in descending order of relevance to the query string being searched. A web page's ranking for a specific query depends on factors like- its relevance to the words and concepts in the query, its overall link popularity etc. There are two categories of these algorithms viz. text based and link based [6].

3.1 Text-Based Ranking

The ranking scheme used in the conventional search engines is purely Text-Based i.e. the pages are ranked based on their textual content, which seems to be logical. In such schemes, the factors that influence the rank of a page are [6]:

- *Number of matched terms* with the query string.
- *Location Factors* influence the rank of a page depending upon where the search string is located on that page. The search query string could be found in the title of a page or in the leading paragraphs of a page or even near the head of a page [6].
- *Frequency Factors* deal with the number of times the search string appears in the page. The more time the string appears, the better is the page ranking [6].

Most of the times, the affect of these factors is considered collectively. For example, if a search string repeatedly appears near the beginning of a page then that page should have a high rank [6].

3.2 Link-Based Ranking Algorithms

Another popular class of ranking algorithms is the link-based algorithms. They view the web as a directed graph where the web pages form the nodes and the hyperlinks between the web pages form the directed edges between these nodes [6]. Link-based ranking algorithms propagate page importance through links. During 1997-1998, two most influential hyperlink based search algorithms were reported. These algorithms are:

- HITS (Hyperlink Induced Topic Search)
- PageRanking algorithm

Both algorithms are related to **social networks**. They exploit the hyperlinks of the Web to rank pages according to their levels of "prestige" or "authority". Section 4 and 5 next individually discuss the above algorithms.

4. HITS Algorithm

4.1 Overview

Hypertext Induced Topic Search (HITS) or hubs and authorities is a link analysis algorithm developed by Jon Kleinberg in 1998 to rate Web pages. A precursor to PageRank, HITS is a search query dependent algorithm that ranks the web page by processing its entire in links and out links [4]. Thus, ranking of the web page is decided by analyzing its textual contents against a given query. When the user issues a search query, HITS first expands the list of relevant pages returned by a search engine and then produces two rankings of the expanded set of pages, authority ranking and hub ranking. In this algorithm a web page is named as authority if the web page is pointed by many hyper links and a web page is named as HUB if the page point to various hyperlinks [4]. The algorithm produces two types of pages:

- *Authority:* pages that provide an important, trustworthy information on a given topic
- *Hub:* pages that contain links to authorities

Figure 3 [8] below depicts the hubs and authorities created by HITS. Authorities and hubs exhibit a mutually reinforcing relationship: a better hub points to many good authorities, and a better authority is pointed to by many good hubs.

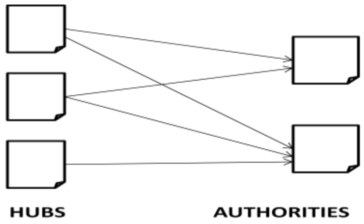


Fig 3: Hubs and Authorities

To mark a web page as Authority or Update, HITS follows the following rules [8, 12]:

Authority Update Rule: ∀p, update auth (p) as follows:

$$\sum_{i=1}^{n} hub(i) \qquad (1)$$

Where n is the total number of pages connected to p. According to (1) the Authority score of a page is the sum of all the Hub scores of pages that point to it [8].

Hub Update Rule: ∀p, we update hub (p) as follows:

$$\sum_{i=1}^{n} auth(i) \qquad (2)$$

Where n is the total number of pages, p connects to. According to (2) a page's Hub score is the sum of the Authority scores of all its linking pages [8].

More precisely, given a set of web pages (say, retrieved in response to a search query), the HITS algorithm first forms the n by n adjacency matrix A, whose m(i , j) element is 1 if page *i* links to page *j* and 0 otherwise.

Adjacency Matrix *A*

$m_{(i,j)}$ = 1 if (i,j) exists in graph ,

$m_{(i,j)}$ = 0 otherwise.

It then iterates the following equations [9]: For each $m_{i,}$

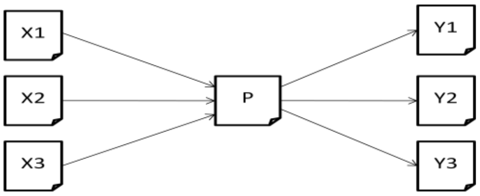
$$a_i^{(t+1)} = \sum_{\{j:j\to i\}} h_j^{(t)}; \qquad (3)$$

$$h_i^{(t+1)} = \sum_{\{j:i\to j\}} a_j^{(t+1)} \qquad (4)$$

(Where "i → j" means page i links to page j and a_i is authority of ith page and h_i is the hub representation of ith page). Figure 4[4] shows an illustration of HITS process.

Normalization:

The final hub-authority scores of nodes are determined after infinite repetitions of the algorithm. On applying the hub update rule and authority update rule directly and iteratively diverging values are obtained. So, it is necessary to normalize the matrix after each iteration [11].



$A_p = H_{X1}+H_{X2}+H_{X3}$ $\qquad$ $H_p = A_{Y1}+A_{Y2}+A_{Y3}$

Figure 4: Illustration of HITS process

4.2 Implementation of HITS algorithm

In the first step of the HITS algorithm the root set (most relevant pages to the query) can be obtained by taking the top n pages returned by a text-based search algorithm. A base set is generated by augmenting the root set with all the web pages that are linked from it and some of the pages that link to it. The web pages in the base set and all hyperlinks among those pages form a focused subgraph. The HITS computation is performed only on this focused subgraph [24]. According to Kleinberg [25], the reason for constructing a base set is to ensure that most (or many) of the strongest authorities are included. The Hub score and Authority score for a node is calculated with the following algorithm [11]:

- Start with each node having a hub score and authority score of 1.
- Run the Authority Update Rule
- Run the Hub Update Rule
- Normalize the values by dividing each Hub score by the sum of the squares of all Hub scores, and dividing each Authority score by the sum of the squares of all Authority scores.
- Repeat from the second step as necessary.

Pseudocode of HITS algorithm [26]

```
Let G be set of pages
for each page pg in G do
      pg.auth = 1          // authority score of the page pg
      pg.hub = 1           // hub score of the page pg
function Calc_Hubs_Authorities(G)
  for step from 1 to i do // run the algorithm for i steps
        norm = 0
        for each page pg in G do // update authority values
              pg.auth = 0
          for each page qg in p.inNeighbors do //set of   pages that       link to pg
                      pg.auth += qg.hub
norm += square(pg.auth) //sum of the                    squared auth values to normalise
         norm = sqrt(normal)
        for each page pg in G do    // update the auth scores
pg.auth = pg.auth / normal // normalise the auth             values
        norm = 0
        for each page pg in G do  //  update hub values
             pg.hub = 0
           for each page rg in pg.outNeighbors do // set of pages that pg links to
                   pg.hub += rg.auth
            norm += square(pg.hub) //sum of the squared hub values to normalise
       norm = sqrt(normal)
       for each page pg in G do  //update hub values
     pg.hub = pg.hub / normal  // normalise the hub values
```

The hub and authority values converge in the pseudocode above. One way to get around this, however, would be to normalize the hub and authority values after each "step" by dividing each authority value by the square root of the sum of the squares of all authority values, and dividing each hub value by the square root of the sum of the squares of all hub values. This is what the pseudocode above does.

4.3 Simulation

4.3.1 Graph case study 1

The Graph shown in the figure 4 shows A, B, C three pages in a small network, which are linked using directed edges. The simulation implementing HITS algorithm on the graph is shown in the figure 5:

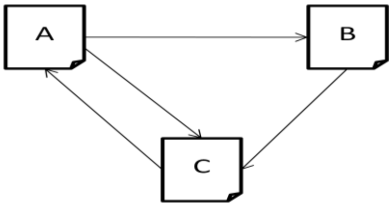
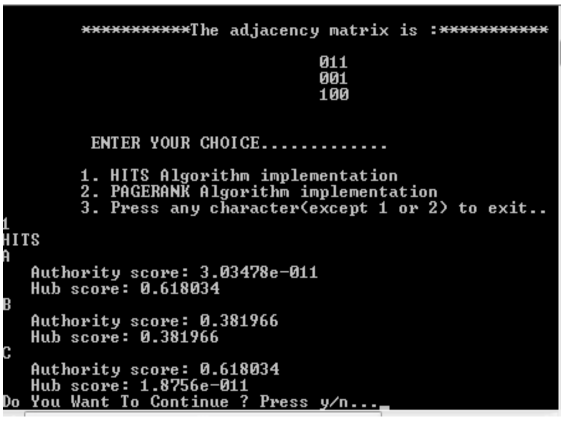


Figure 5: A connected graph

Figure 6 depicts the hubs and authority scores for each node calculated using simulation.



```
         ***********The adjacency matrix is :***********

                                   011
                                   001
                                   100


          ENTER YOUR CHOICE............

         1. HITS Algorithm implementation
         2. PAGERANK Algorithm implementation
         3. Press any character(except 1 or 2) to exit..
1
HITS
A
   Authority score: 3.03478e-011
   Hub score: 0.618034
B
   Authority score: 0.381966
   Hub score: 0.381966
C
   Authority score: 0.618034
   Hub score: 1.8756e-011
Do You Want To Continue ? Press y/n...
```

Figure 6: Simulation for graph1

4.3.2 Graph case study 2

The Graph in figure 7 represents A, B, C, D, four pages in a small network, which are linked through directed edges. The simulation implementing HITS algorithm on the graph is shown in the figure 8:

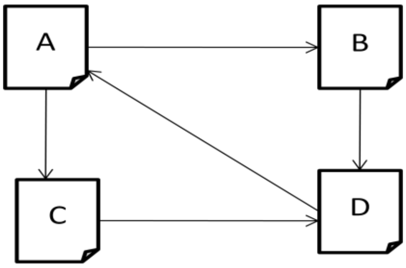
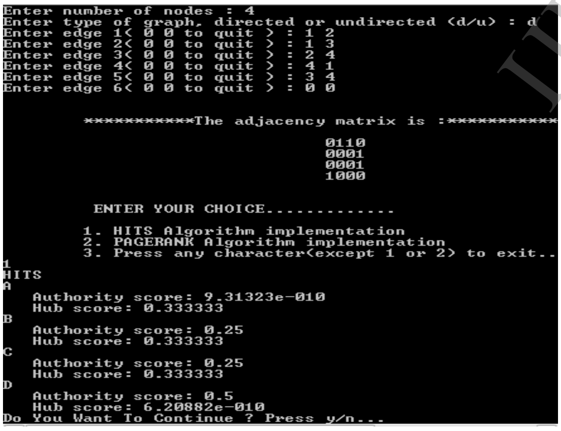


Figure 7: A connected graph

The figure 8 shows the hubs and authority scores for each node calculated using simulation.



```
Enter number of nodes : 4
Enter type of graph, directed or undirected (d/u) : d
Enter edge 1( 0 0 to quit ) : 1 2
Enter edge 2( 0 0 to quit ) : 1 3
Enter edge 3( 0 0 to quit ) : 2 4
Enter edge 4( 0 0 to quit ) : 4 1
Enter edge 5( 0 0 to quit ) : 3 4
Enter edge 6( 0 0 to quit ) : 0 0

        ***********The adjacency matrix is :***********

                                0110
                                0001
                                0001
                                1000


         ENTER YOUR CHOICE.............

        1. HITS Algorithm implementation
        2. PAGERANK Algorithm implementation
        3. Press any character(except 1 or 2) to exit..
1
HITS
A
   Authority score: 9.31323e-010
   Hub score: 0.333333
B
   Authority score: 0.25
   Hub score: 0.333333
C
   Authority score: 0.25
   Hub score: 0.333333
D
   Authority score: 0.5
   Hub score: 6.20882e-010
Do You Want To Continue ? Press y/n...
```

Figure 8: Simulation for graph2

4.3.3 Graph case study 3

The Graph in figure 9 represents a more complex network of 7 pages.

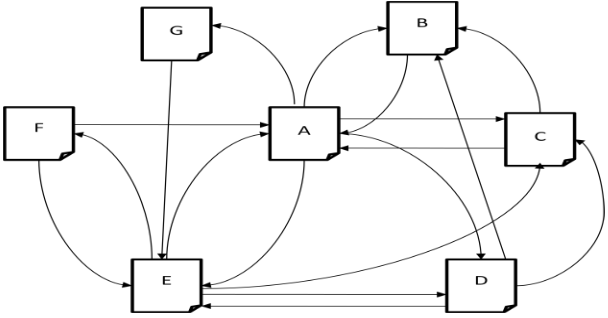
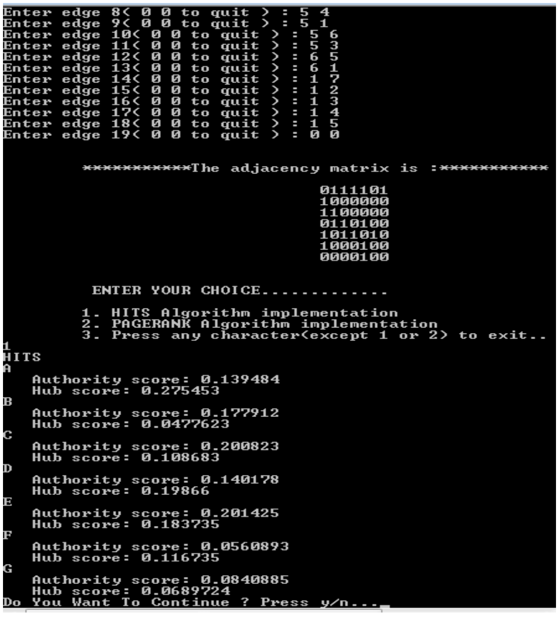


Fig 9: A connected graph

Figure 10 below illustrates the hubs and authority scores for each node calculated using simulation.



```
Enter edge 8< 0 0 to quit > : 5 4
Enter edge 9< 0 0 to quit > : 5 1
Enter edge 10< 0 0 to quit > : 5 6
Enter edge 11< 0 0 to quit > : 5 3
Enter edge 12< 0 0 to quit > : 6 5
Enter edge 13< 0 0 to quit > : 6 1
Enter edge 14< 0 0 to quit > : 1 7
Enter edge 15< 0 0 to quit > : 1 2
Enter edge 16< 0 0 to quit > : 1 3
Enter edge 17< 0 0 to quit > : 1 4
Enter edge 18< 0 0 to quit > : 1 5
Enter edge 19< 0 0 to quit > : 0 0

        ***********The adjacency matrix is :***********

                                0111101
                                1000000
                                1100000
                                0110100
                                1011010
                                1000100
                                0000100

         ENTER YOUR CHOICE.............

        1. HITS Algorithm implementation
        2. PAGERANK Algorithm implementation
        3. Press any character(except 1 or 2) to exit..
1
HITS
A
   Authority score: 0.139484
   Hub score: 0.275453
B
   Authority score: 0.177912
   Hub score: 0.0477623
C
   Authority score: 0.200823
   Hub score: 0.108683
D
   Authority score: 0.140178
   Hub score: 0.19866
E
   Authority score: 0.201425
   Hub score: 0.183735
F
   Authority score: 0.0560893
   Hub score: 0.116735
G
   Authority score: 0.0840885
   Hub score: 0.0689724
Do You Want To Continue ? Press y/n...
```

Figure 10: Simulation for graph3

4.4 Advantages of HITS

We list below a few considerable advantages of HOTS:

1. HITS scores due to its ability to rank pages according to the query string, resulting in relevant authority and hub pages.
2. The ranking may also be combined with other information retrieval based rankings.
3. HITS is sensitive to user query (as compared to PageRank).
4. Important pages are obtained on basis of calculated authority and hubs value.
5. HITS is a general algorithm for calculating authority and hubs in order to rank the retrieved data.
6. HITS induces Web graph by finding set of pages with a search on a given query string.
7. Results demonstrates that HITS calculates authority nodes and hubness correctly.

4.5 Drawbacks of HITS algorithm

Some notable drawbacks of HITS algorithm are:

1. **Query Time cost:** The query time evaluation is expensive. This is a major drawback since HITS is a query dependent algorithm.
2. **Irrelevant authorities:** The rating or scores of authorities and hubs could rise due to flaws done by the web page designer. HITS assumes that when a user creates a web page he links a hyperlink from his page to another authority page, as he honestly believes that the authority page is in some way related to his page (hub).
3. **Irrelevant Hubs:** A situation may occur when a page that contains links to a large number of separate topics may receive a high hub rank which is not relevant to the given query. Though this page is not the most relevant source for any information, it still has a very high hub rank if it points to highly ranked authorities.
4. **Mutually reinforcing relationships between hosts:** HITS emphasizes mutual reinforcement between authority and hub webpages. A good hub is a page that points to many good authorities and a good authority is a page that is pointed to by many good hubs.
5. **Topic Drift:** Topic drift occurs when there are irrelevant pages in the root set and they are strongly connected. Since the root set itself contains non-relevant pages, this will reflect on to the pages in the base set. Also, the web graph constructed from the pages in the base set, will not have the most relevant nodes and as a result the algorithm will not be able to find the highest ranked authorities and hubs for a given query.
6. **Less Feasibility:** HITS invokes a traditional search engine to obtain a set of pages relevant to it, expands this set with its inlinks and outlinks, and then attempts to find two types of pages, *hubs* (pages that point to many pages of high quality) and *authorities* (pages of high quality)[20]. Because this

computation is carried out at query time, it is not feasible for today's search engines, which need to handle tens of millions of queries per day [20].

5. PageRank Algorithm

5.1 Overview

The PageRank algorithm originally developed at Stanford University by Larry Page and Sergey Brin in 1996 as part of a research project about a new search engine. PageRank is a link analysis algorithm, named after Larry Page and used by the Google Internet search engine. The algorithm assigns a numerical weight or rank to each page of a hyperlinked set of documents with the purpose of measuring its relative importance within the set. The Page Rank algorithm utilizes link structure of the web pages. This algorithm is query independent and it operates on the whole Web and assigns a PageRank to every web page [13]. It is based on the concepts that if a page contains important links towards it then the links of this page towards the other page are also to be considered as important pages i.e. if an authoritative web page A links to page B, then B is also authoritative. Page Rank utilizes the back link in deciding the rank score. If the summation of all the ranks of the back links is large then the page then it is provided a large rank [4]. A simplified version of PageRank is also suggested in [4]:

$$PR(u) = \sum_{v \epsilon Bu} \frac{PR(v)}{L(v)} \qquad (5)$$

In (5) PageRank value for a web page u is dependent on the PageRank values for each web page v out of the set Bu (this set contains all pages linking to web page u), divided by the number L(v) of links from page v. An illustration for back links among set of pages is shown in the diagram in figure 11. Here B is the back link of A, D and A, B, E are back links for C and D is the back link for E.

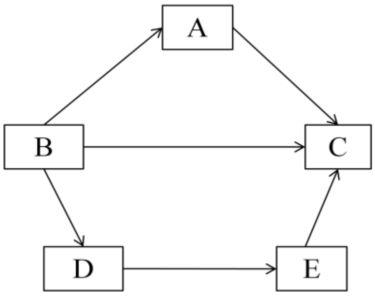


Fig 11: An illustration of back links

Ranking of web pages by the Google search engine was initially determined by three factors [18]:

- Page specific factors
- Anchor text of inbound links
- PageRank

Page specific factors may constitute body text, HTML-tag weight component (e.g. title preference), the URL of the document etc. Many other factors have also joined the ranking methods of the Google search engine. To provide search results, Google computes an IR score out of page specific factors and the anchor text of inbound links of a page, which is weighted by position and accentuation of the search term within the document [18]. The IR-score is then multiplied with PageRank as an indicator for the general importance of the page [18].

5.2 Implementation of PageRank algorithm

The PageRank algorithm does not rank the whole website, but it's determined for each page individually. Furthermore, the PageRank of page A is recursively defined by the PageRank of those pages which link to page A. PageRank is a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. Thus, a PageRank of 0.5 means there is a 50% chance that a person clicking on a random link will be directed to the document with the 0.5 PageRank. Brin S. and L. Page described PageRank formula as below [10]:

PR(A)=(1-d)+d(PR(T1)/C(T1)+………+ PR(Tn)/C(Tn)) (6)

Where:
PR(A)= PageRank of page A

T1….Tn=All pages that link to page A
PR(Ti)=Page rank of page Ti
C(Ti)=the number of pages to which Ti links to
d= damping factor which can be set between 0 and 1
PR(Ti)/C(Ti)= PageRank of Ti distributing to all pages that Ti links to.
(1-d)= To make up for some pages that do not have any out-links to avoid losing some page ranks.
Each additional inbound link for a web page always increases that page's PageRank[18]. One may assume that an additional inbound link from page X increases the PageRank of page A by [18]: d × PR(X) / C(X)

Damping factor: The PageRank theory holds that any imaginary surfer who is randomly clicking on links will eventually stop clicking. The probability, at any step, that the person will continue is called a damping factor *d*. The damping factor can be set to any value such that 0<d<1, nominally it is set around 0.85. The damping factor is subtracted from 1 and this term is then added to the product of the damping factor and the sum of the incoming PageRank scores.

Pseudocode for PageRank (G) [6]
Input: Let G represent set of nodes or web pages
Output: An n-element array of PR which represent PageRank for each web page

```
For i ← 0 to n-1 do
        Let A be an array of n elements
         A[i] ← 1/n
d ← some value 0<d<1, e.g. 0.15, 0.85
Repeat
    For i ← 0 to n-1 do
        Let PR be a n-element of array
        PR[i] ← 1-d
        For all pages Q such that Q links to PR[i] do
         Let $O_n$ be the number of outgoing edge of Q
              PR[i] ← PR[i]+ d * A[Q]/$O_n$
    If the difference between J and PR is small do
        Return PR
    For i ← 0 to n-1 do
        A[i] ← PR[i]
```

5.2 Simulation

5.3.1 Graph case study 1

The graph in figure 12[14] shows how 3 pages are linked in a network. Inside each page is shown its calculated PageRank. The illustration demonstrates the application of PageRank in a simplified 3 page internet [14].

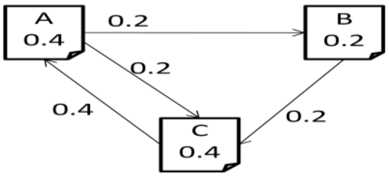
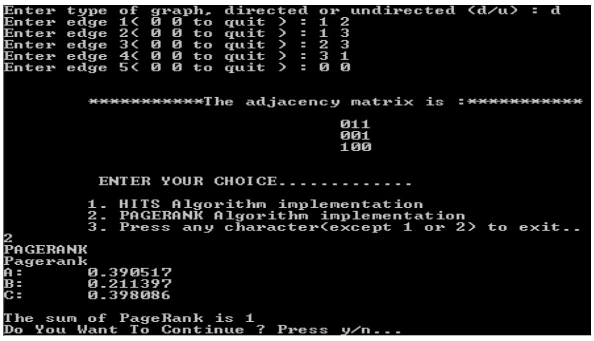


Fig 12: PageRank in a simplified 3 page internet.

The figure 13 below shows the PageRank for each page calculated by simulation.



```
Enter type of graph, directed or undirected (d/u) : d
Enter edge 1( 0 0 to quit ) : 1 2
Enter edge 2( 0 0 to quit ) : 1 3
Enter edge 3( 0 0 to quit ) : 2 3
Enter edge 4( 0 0 to quit ) : 3 1
Enter edge 5( 0 0 to quit ) : 0 0

        ***********The adjacency matrix is :***********

                                011
                                001
                                100

         ENTER YOUR CHOICE.............

        1. HITS Algorithm implementation
        2. PAGERANK Algorithm implementation
        3. Press any character(except 1 or 2) to exit..
2
PAGERANK
Pagerank
A:      0.390517
B:      0.211397
C:      0.398086

The sum of PageRank is 1
Do You Want To Continue ? Press y/n...
```

Fig 13: Simulation for graph1

5.3.2 Graph case study 2

The graph in figure 14 illustrates a model of 4 nodes for demonstrating the PageRank.

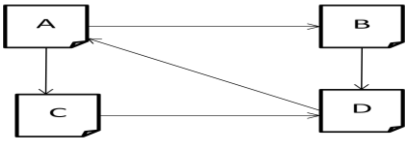
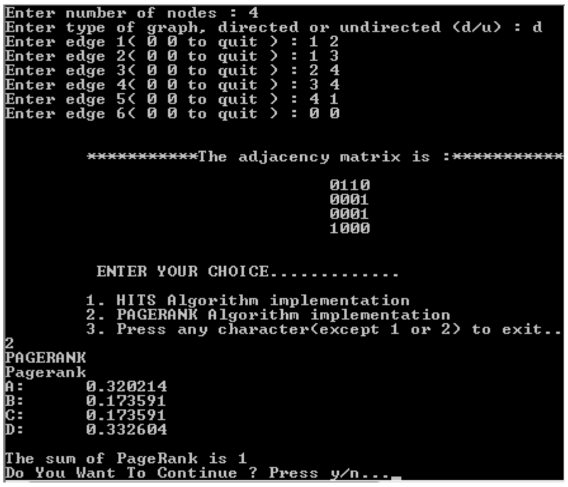


Fig 14: Graph having 4 nodes

Below the figure 15 illustrates the simulation output for graph 2.



```
Enter number of nodes : 4
Enter type of graph, directed or undirected (d/u) : d
Enter edge 1( 0 0 to quit ) : 1 2
Enter edge 2( 0 0 to quit ) : 1 3
Enter edge 3( 0 0 to quit ) : 2 4
Enter edge 4( 0 0 to quit ) : 3 4
Enter edge 5( 0 0 to quit ) : 4 1
Enter edge 6( 0 0 to quit ) : 0 0


        ***********The adjacency matrix is :***********

                                0110
                                0001
                                0001
                                1000


         ENTER YOUR CHOICE.............

        1. HITS Algorithm implementation
        2. PAGERANK Algorithm implementation
        3. Press any character(except 1 or 2) to exit..
2
PAGERANK
Pagerank
A:      0.320214
B:      0.173591
C:      0.173591
D:      0.332604

The sum of PageRank is 1
Do You Want To Continue ? Press y/n...
```

Fig 15: Simulation for Graph2

5.3.3 Graph case study 3

The graph in figure 16 [14] illustrates a model of greater complexity which more accurately demonstrates the functionality of PageRank [14].

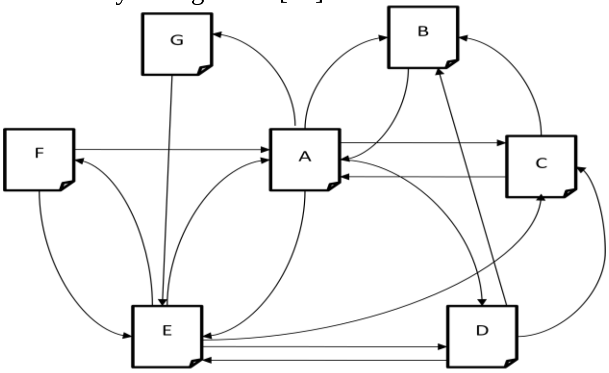
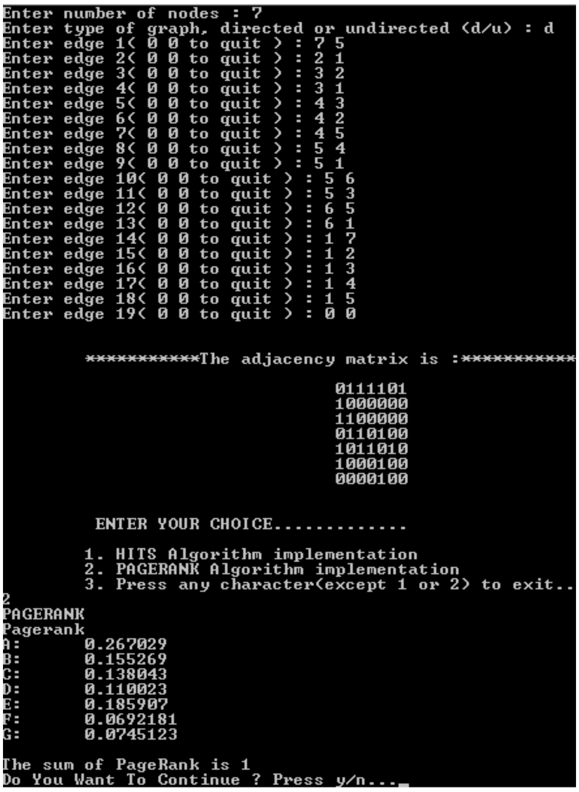


Fig 16: Graph of a complex network internet.

Figure 17 below shows the PageRank for each page calculated by simulation.



```
Enter number of nodes : 7
Enter type of graph, directed or undirected (d/u) : d
Enter edge 1( 0 0 to quit ) : 7 5
Enter edge 2( 0 0 to quit ) : 2 1
Enter edge 3( 0 0 to quit ) : 3 2
Enter edge 4( 0 0 to quit ) : 3 1
Enter edge 5( 0 0 to quit ) : 4 3
Enter edge 6( 0 0 to quit ) : 4 2
Enter edge 7( 0 0 to quit ) : 4 5
Enter edge 8( 0 0 to quit ) : 5 4
Enter edge 9( 0 0 to quit ) : 5 1
Enter edge 10( 0 0 to quit ) : 5 6
Enter edge 11( 0 0 to quit ) : 5 3
Enter edge 12( 0 0 to quit ) : 6 5
Enter edge 13( 0 0 to quit ) : 6 1
Enter edge 14( 0 0 to quit ) : 1 7
Enter edge 15( 0 0 to quit ) : 1 2
Enter edge 16( 0 0 to quit ) : 1 3
Enter edge 17( 0 0 to quit ) : 1 4
Enter edge 18( 0 0 to quit ) : 1 5
Enter edge 19( 0 0 to quit ) : 0 0


        ***********The adjacency matrix is :***********

                                0111101
                                1000000
                                1100000
                                0110100
                                1011010
                                1000100
                                0000100


         ENTER YOUR CHOICE.............

        1. HITS Algorithm implementation
        2. PAGERANK Algorithm implementation
        3. Press any character(except 1 or 2) to exit..
2
PAGERANK
Pagerank
A:      0.267029
B:      0.155269
C:      0.138043
D:      0.110023
E:      0.185907
F:      0.0692181
G:      0.0745123

The sum of PageRank is 1
Do You Want To Continue ? Press y/n...
```

Fig 17: PageRank calculated using simulation for graph3.

5.4 Complexity Analysis

On observing the pseudocode for PageRank algorithm as in section 5.2 the running time of the algorithm is depends on three factors: number of iterations (i), number of web pages (n) and number of outgoing edges of each web page (O_n) [6]. The complexity is roughly

$$n + in\sum_{i=1}^{n} o_i + n = n\,(2 + i\sum_{i=1}^{n} o_i\,) \qquad (6)$$

Since the number of iterations and the outgoing edges of each page is pretty small compare to number of web pages. The complexity of the PageRank is O(n) [6].

5.5 Strengths of PageRank algorithm

The strengths of PageRank algorithm are as follows:

1. **Less Query time cost:** PageRank has a clear advantage over the HITS algorithm, as the query-time cost of incorporating the precomputed PageRank importance score for a page is low [19].
2. **Less susceptibility to localized links:** Furthermore, as PageRank is generated using the entire Web graph, rather than a small subset, it is less susceptible to localized link spam[19].
3. **More Efficient:** In contrast, PageRank computes a single measure of quality for a page at crawl time. This measure is then combined with a traditional information retrieval score at query time. Compared with HITS, this has the advantage of much greater efficiency [20].
4. **Feasibility:** As compared to Hits algorithm the PageRank algorithm is more feasible in today's scenario since it performs computations at crawl time rather than query time.

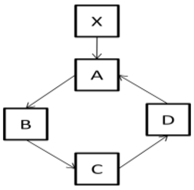


Figure 19: Illustration of circular references

5.6 Drawbacks of PageRank algorithm

The following are the problems or disadvantages[17] of PageRank:

1. **Rank Sinks**: The Rank sinks problem occurs when in a network pages get in infinite link cycles as shown in the figure 18 [17] below:

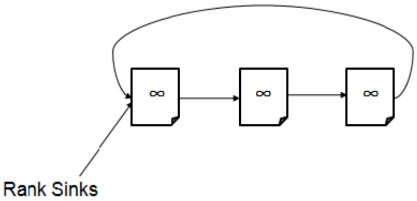


Figure 18: Illustration of Rank Sinks

2. **Spider Traps**: Another problem in PageRank is Spider Traps. A group of pages is a spider trap if there are no links from within the group to outside the group.
3. **Dangling Links**: This occurs when a page contains a link such that the hypertext points to a page with no outgoing links. Such a link is known as Dangling Link.
4. **Dead Ends**: Dead Ends are simply pages with no outgoing links.

5. PageRank doesn't handle pages with no outedges very well, because they decrease the PageRank overall.
6. **Circular References** : If you have circle references in your website, then it will reduce your front page's PageRank [18]. The figure 19 [18] shown below illustrates the case of circular references.

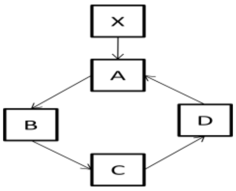


Figure 19: Illustration of circular references

7. **Effect of additional pages:** If you add a web page to your website it will increase your page's rank by ≈0.428 [18]. The problem with this method is that if you increase your front page's PageRank by adding additional pages, than the rank of your other pages will go down[18]. The solution is to swap links with websites which have high PageRank value. The easiest way to do this is to make a page with high PageRank and link it to your front page[18].
8. PageRank score of a page ignores whether or not the page is relevant to the query at hand.

6. Variations of PageRank Algorithm

6.1 Second Page Rank algorithm

A second Page Rank algorithm [14] was published by Lawrence Page and Sergey Brin. This second algorithm does not differ significantly from the original, it does however offer a better explanation of the "random surfer" model, which justifies PageRank by stating its effectiveness in mapping the probability that a random surfer will wind up on any given page [14]. The random surfer visits a page according to a certain probability, which is the PageRank of that page.

Example:

Page A has a PageRank of 25, and there are 5,000,000,000 pages on the Internet. It would then follow that there is a 25 to 5,000,000,000 chance that "random surfer" is viewing Page A right now. The Second Algorithm [14]:

$$\mathbf{PR(A) = (1-d) / N + d\ (PR(T1)/C(T1) + \ldots + PR(Tn)/C(Tn))}$$

N has been introduced to represent the total number of pages on the WWW. This algorithm forms a clearer probability distribution by asserting the number of variables that random surfer could encounter [14].

6.2 Weighted Page Rank Algorithm

Weighted Page Rank [15] Algorithm is proposed by WenpuXing and Ali Ghorbani. Weighted page rank algorithm (WPR) is a modification of the original page rank algorithm. The rank scores are decided based on the popularity of the pages by taking into consideration the importance of both the in-links and out-links of the pages. This algorithm does not equally divide the rank of a page among its out-link pages and provides high value of rank to the more popular pages. The rank value is given to every out-link page based on its popularity. Popularity of a page is decided by observing its number of in links and out links [4].

6.3 Weighted Links Rank Algorithm

This algorithm represents a modification of the standard page rank algorithm and is given by Ricardo Baeza-Yates and Emilio Davis named as weighted links rank (WLRank). This algorithm follows the techniques of Web Structure Mining and Web Content Mining. The algorithm suggests the modification that it provides weight value to the link based on three major parameters i.e. length of the anchor text, tag in which the link is contained and relative position in the page [4]. The best attribute for providing weight value to the link is the length of anchor. Relative position was not so effective, indicating that the logical position not always matches the physical position. Future work in this algorithm includes, tuning of the weight factor of every term for further evolution [4].

6.4 PageRank for undirected graphs

Although traditionally applied on directed graphs, recursive graph-based ranking algorithms can be also applied to undirected graphs, in which case the out degree of a vertex is equal to the in-degree of the vertex [21]. For loosely connected graphs, with the number of edges proportional with the number of vertices, undirected graphs tend to have more gradual convergence curves. As the connectivity of the graph increases (i.e. larger number of edges), convergence is usually achieved after fewer iterations, and the convergence curves for directed and undirected graphs practically overlap [21]. The PageRank of an undirected graph G is statistically close to the degree distribution of the graph G, but they are generally not identical: If R is the PageRank vector defined above, and D is the degree distribution vector:

$$D = \frac{1}{2E}\begin{pmatrix} \deg(p_1) \\ \vdots \\ \deg(p_n) \end{pmatrix} \qquad (7)$$

where deg(p_i)denotes the degree of vertex p_i, and E is the edge-set of the graph, then, the PageRank of an undirected graph equals to the degree distribution vector if and only if the graph is regular, i.e., every vertex has the same degree.

7. Comparison of HITS and PageRank.

Table 1 below enlists the comparison of HITS and PageRank algorithm.

Table 1. Comparison of HITS and PageRank algorithms

Criteria	HITS	Page Rank
Basic Criteria	Link analysis algorithm	Link analysis algorithm based on random surfer model.
Main Technique followed	Web Structure Mining, Web Content Mining	Web Structure Mining
Efficiency	For a given a query HITS invokes traditional search engine to retrieve set of pages relevant to it and then attempts to find hubs and authorities. Since this computation is carried out at query time, it is not feasible for today's search engines, which need to handle millions of queries per day.	PageRank computes a single measure of quality for a page at crawl time. This measure is then combined with a traditional information retrieval score at query time. The advantage is much greater efficiency
Mutual Reinforcement	HITS emphasizes mutual reinforcement between authority and hub webpages	PageRank does not attempt to capture the distinction between hubs and authorities. It ranks pages just by

		authority.
Neighborhood	HITS is applied to the local neighborhood of pages surrounding the results of a query	PageRank is applied to the entire web
Query Dependency	HITS is query dependent	PageRank is query-independent
Stability	Can be unstable: changing a few links can lead to quite different rankings.	Can be unstable: changing a few links can lead to quite different rankings.
Input Parameter(s)	Content, Back and Forward links	Back links
Analysis Scope	Single Page	Single Page
Relevancy	Less. Since this algorithm ranks the pages on the indexing time	More since this algorithm uses the hyperlinks to give good results and also consider the content of the page
Quality of Results obtained	Less than PageRank algorithm	Medium
Complexity Analysis	$O(kN^2)$	O(n)
Merits	→Hub and Authority values are calculated so that the relevant and important pages are obtained. →HITS is a general algorithm used for calculating the authority and hubs in order to rank the retrieved data →The basic aim of that algorithm is to induce the Web graph by finding set of pages with a search on a given topic (query). →Results demonstrates that it is good in calculating the authority nodes and hubness.	→Used in journal citations and in academics → Google technology for ranking web pages. →Query-time cost of incorporating precomputed PageRank importance score for a page is low → PageRank generated using the entire Web graph, rather than a small subset, it is Less susceptible to localized link spam. → PageRank may be used as a methodology to measure the impact of a community like the blogosphere on the overall Web itself.
Limitations	→Query Dependency →Irrelevant authorities problem →Irrelevant Hubs problem →Mutually reinforcing relationships between hosts problematic →Topic Drift	→Rank Sinks →Spider Traps →Dangling Links →Dead Ends →Circular References →Effect of additional pages

8. Conclusion

On the basis of this study we conclude that both page rank and HITS algorithm are different link analysis algorithms that employ different models to calculate web page rank. Page Rank is a more popular algorithm used as the basis for the very popular Google search engine. This popularity is due to the features like efficiency, feasibility, less query time cost, less susceptibility to localized links etc. which are absent in HITS algorithm. However though the HITS algorithm itself has not been very popular, different extensions of the same have been employed in a number of different web sites.

9. References

[1] Comparative Study of Web 1.0, Web 2.0 and Web 3.0, Umesha Naik D Shivalingaiah

[2] Technivision Knowledge Base **Search Engines** A Brief Overview of How They Work In Everyday English! January 1, 2009 Prepared by: Kevin MacDonald

[3] A Novel Architecture of Ontology-based Semantic Web Crawler, Ram Kumar Rana IIMT Institute of Engg. & Technology, Meerut, India ,Nidhi Tyagi Shobhit University, Meerut, India

[4] A Comparative Analysis of Web Page Ranking Algorithms, Dilip Kumar Sharma et al. / (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 08, 2010, 2670-2676

[5] Ranking Techniques for Social Networking Sites based on Popularity, Mercy Paul Selvan et al / Indian Journal of Computer Science and Engineering (IJCSE).

[6] World Wide Web searching technique, Vineel Katipally, Leong-Chiang Tee, Yang Yang Computer Science & Engineering Department Arizona State University

[7] Cho, J.; Adams, R.E.; Page quality: In search of an unbiased web ranking, Technical report, UCLA Computer Science Department, November 2003.

[8] "Survey on Web Page Ranking Algorithms", Mercy Paul Selvan, A .Chandra Sekar, A.Priya Dharshin *International Journal of Computer Applications (0975 – 8887) Volume 41– No.19, March 2012*

[9] Stable Algorithms for Link Analysis By: Andrew Y. Ng, Alice X. Zheng, Michael I. Jordan CS Div. & Dept. of Stat. U.C. Berkeley.

[10] PageRank explained *or "Everything you've always wanted to know about PageRank"* Written and theorised by Chris Ridings.

[11] Association Rule Mining based on Ontological Relational Weights, N. Radhika, K.Vidya, Department of Computer Science and Engineering, Aurora's Technological and Research Institute, India.

[12] Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins, *The Web as a graph: measurements, models and methods*, Proc. Fifth Ann. Int. Computing and Combinatorics Conf., Springer-Verlag Lecture Notes in Computer Science 1627, 1999, 1-17.

[13] "Finding Authorities and Hubs From Link Structures on the World Wide Web" by-Allan Borodin, Gareth O. Roberts, Jeffrey S. Rosenthal, and Panayiotis Tsaparas

[14] "Google PageRank™ and Related Technologies" by Jason J. Green

[15] Wenpu Xing and Ali Ghorbani, "Weighted PageRank Algorithm", In proceedings of the 2rd Annual Conference on Communication Networks & Services Research, PP. 305-314, 2004.

[16] "A Syntactic Classification based Web Page Ranking Algorithm", Debajyoti Mukhopadhyay , Pradipta Biswas , Young-Chon Kim

[17] "Modeling and Optimizing Hypertextual Search Engines" Based on the Reasearch of Larry Page and Sergey Brin, Yunfei Zhao Department of Computer Science, University of Vermont Slides from Spring 2009 Presenter: Michael Karpeles

[18] Google and the Page Rank Algorithm, slides by Székely Endre 2007. 01. 18.

[19] Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search Taher H. Haveliwala Stanford University taherh@cs.stanford.edu

[20] The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank Matthew Richardson Pedro Domingos Department of Computer Science and Engineering University of Washington Box 352350 Seattle, WA 98195-2350, USA *{mattr, pedrod}@cs.washington.edu*

[21] Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization Rada Mihalcea Department of Computer Science University of North Texas rada@cs.unt.edu

[22] Hyperlink Analysis: Techniques and Applications Prasanna Desikan, Jaideep Srivastava, Vipin Kumar, and Pang-Ning Tan Department of Computer Science, University of Minnesota, Minneapolis, MN, USA {desikan, srivastava, kumar, ptan} @cs.umn.edu

[23] J. Srivastava, R. Cooley, M. Deshpande, and P. –N. Tan. "Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data" (2000), SIGKDD Explorations, Vol. 1, Issue 2, 2000.

[24] HAR: Hub, Authority and Relevance Scores in Multi-Relational Data for Query Search, Xutao Li, Michael K. Ng, Yunming Ye

[25] Authoritative Sources in a Hyperlinked Environment, JON M. KLEINBERG *Cornell University, Ithaca, New York*

[26] Analysis of data mining techniques for increasing search speed in web, B.Chaitanya Krishna, C.Niveditha, G.Anusha , U.Sindhu , Sk.Silar.