ISSN:2229-6093

COMPARATIVE STUDY OF PAGE RANK ALGORITHM WITH DIFFERENT RANKING ALGORITHMS ADOPTED BY SEARCH ENGINE FOR WEBSITE RANKING

MRIDULA BATRA[(1)] SACHIN SHARMA[(2)]

[1,2] ASSTT PROF, DEPTT OF COMPUTER APPLICATIONS, MANAV RACHNA INTERNATIONAL UNIVERSITY, FARIDABAD

Abstract

We use Search Engines to search for information across the Internet. Internet being an ever-expanding ocean of data, their importance grew with every passing day. The diversity of the information itself made it necessary to have a tool to cut down on the time spent in searching. Page Rank is a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. The main disadvantage is that it favors older pages, because a new page, even a very good one, will not have many links unless it is part of an existing site. Trust Rank is a major factor that now replaces PageRank as the flagship of parameter groups in the Google algorithm. It is of key importance for calculating ranking positions and the crawling frequency of web sites. Page Rank (Google PR) and Trust Rank are the two main issues which are discussed frequently on various SEO forums. In this paper, we will compare these algorithms.

Keywords: Ranking, Page, Trust, Hits

1.1. Introduction

The web creates new challenges for information retrieval. The amount of information on the web is growing rapidly, as well as the number of new users inexperienced in the art of web research. People are likely to surf the web using its link graph, often starting with high quality human maintained indices such as Yahoo! or with search engines. Human maintained lists cover popular topics effectively but are subjective, expensive to build and maintain, slow to improve, and cannot cover all esoteric topics. Automated search engines that rely on keyword matching usually return too many low quality matches. To make matters worse, some advertisers attempt to gain people's attention by taking measures meant to mislead automated search engines. We have built a large-scale search engine which addresses many of the problems of existing systems. It makes especially heavy use of the additional structure present in hypertext to provide much higher quality search results. We chose our system name, Google, because it is a common spelling of googol, or 10^{100} and fits well with our goal of building very large-scale search engines.

1.2. Google Architecture Overview

In this section, we will give a high level overview of how the whole system works as pictured in Figure 1.

Most of Google is implemented in C or C++ for efficiency and can run in either Solaris or Linux. In Google, the web crawling (downloading of web pages) is done by several distributed crawlers. There is a URL server that sends lists of URLs to be fetched to the crawlers. The web pages that are fetched are then sent to the store server. The store server then compresses and stores the web pages into a repository. Every web page has an associated ID number called a docID which is assigned whenever a new URL is parsed out of a web page. The indexing function is performed by the indexer and the sorter. The indexer performs a number of functions. It reads the repository, uncompresses the documents, and parses them. Each document is converted into a set of word occurrences called hits. The hits record the word, position in document, an approximation of font size, and capitalization. The indexer distributes these hits into a set of "barrels", creating a partially sorted forward index. The indexer performs another important function. It parses out all the links in every web page and stores important information about them in an anchors file. This file contains enough information to determine where each link points from and to, and the text of the link.

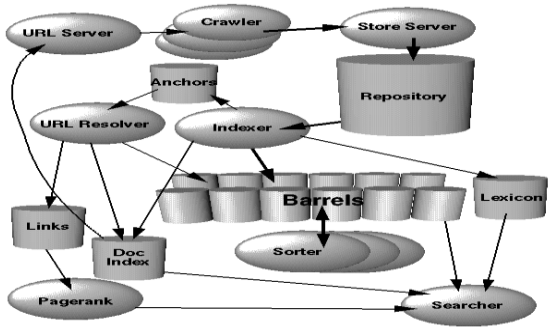


Figure 1: High Level Google Architecture

1.3. Major Data Structures

Google's data structures are optimized so that a large document collection can be crawled, indexed, and searched with little cost.

1.3.1. BigFiles

BigFiles are virtual files spanning multiple file systems and are addressable by 64 bit integers. The allocation among multiple file systems is handled automatically. The BigFiles package also handles allocation and deallocation of file descriptors, since the operating systems do not provide enough for our needs. BigFiles also support rudimentary compression options.

1.3.2. Repository

The repository contains the full HTML of every web page. Each page is compressed using zlib. The choice of compression technique is a tradeoff between speed and compression ratio. We chose zlib's speed over a significant improvement in compression offered by bzip. The compression rate of bzip was approximately 4 to 1 on the repository as compared to zlib's 3 to 1 compression. In the repository, the documents are stored one after the other and are prefixed by docID, length.

1.3.3. Lexicon

The lexicon has several different forms. One important change from earlier systems is that the lexicon can fit in memory for a reasonable price. In the current implementation we can keep the lexicon in memory on a machine with 256 MB of main memory. The current lexicon contains 14 million words (though some rare words were not added to the lexicon).

1.3.4. Hit Lists

A hit list corresponds to a list of occurrences of a particular word in a particular document including position, font, and capitalization information. Hit lists account for most of the space used in both the forward and the inverted indices. Because of this, it is important to represent them as efficiently as possible. We considered several alternatives for encoding position, font, and capitalization -- simple encoding (a triple of integers), a compact encoding (a hand optimized allocation of bits), and Huffman coding.

1.4. Search engine optimization (SEO)

It is the process of improving the volume and quality of traffic to a web site from search engines via "natural" ("organic" or "algorithmic") search results. Usually, the earlier a site is presented in the search results, or the higher it "ranks," the more searchers will visit that site. SEO can also target different kinds of search, including image search, local search, and industry-specific vertical search engines.

1.5. Page Rank: Bringing Order to the Web

The citation (link) graph of the web is an important resource that has largely gone unused in existing web search engines. We have created maps containing as many as 518 million of these hyperlinks, a significant sample of the total. These maps allow rapid calculation of a web page's "Page Rank", an objective measure of its citation importance that corresponds well with people's subjective idea of importance. Because of this correspondence, Page Rank is an excellent way to prioritize the results of web keyword searches. For most popular subjects, a simple text matching search that is restricted to web page titles performs admirably when Page Rank prioritizes the results. For the type of full text searches in the main Google system, Page Rank also helps a great deal.

1.5.1. Description of PageRank Calculation

Academic citation literature has been applied to the web, largely by counting citations or back links to a given page. This gives some approximation of a page's importance or quality. PageRank is defined as follow:

Let us suppose page A has pages T1...Tn which point to it (i.e., are citations). The parameter d is a damping factor which can be set between 0 and 1. We usually set d to 0.85. There are more details about d in the next section. Also C(A) is defined as the number of links going out of page A. The PageRank of a page A is given as follows:

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

Note that the Page Ranks form a probability distribution over web pages, so the sum of all web pages' Page Ranks will be one.

Page Rank or *PR(A)* can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the web. Also, a Page Rank for 26 million web pages can be computed in a few hours on a medium size workstation

1.5.2. ALGORITHM

Page Rank is a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. PageRank can be calculated for any-size collection of documents. It is assumed in several research papers that the distribution is evenly divided between all documents in the collection at the beginning of the computational process. The PageRank computations require several passes, called "iterations", through the collection to adjust approximate PageRank values to more closely reflect the theoretical true value.

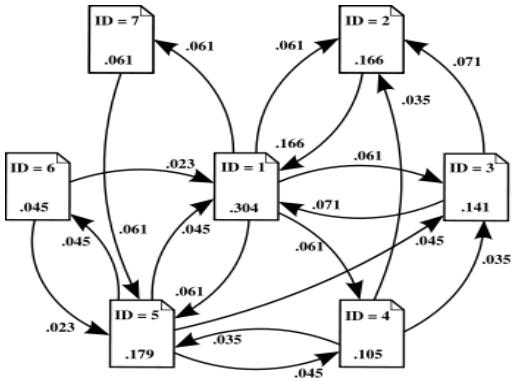


Figure 2

A probability is expressed as a numeric value between 0 and 1. A 0.5 probability is commonly expressed as a "50% chance" of something happening. Hence, 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 Page Rank.

1.5.3. How Page Rank works

Assume a small universe of four web pages: A, B, C and D. The initial approximation of PageRank would be evenly divided between these four documents. Hence, each document would begin with an estimated PageRank of 0.25.

In the original form of PageRank initial values were simply 1. This meant that the sum of all pages was the total number of pages on the web. Later versions of PageRank (see the below formulas) would assume a probability distribution between 0 and 1. Here we're going to simply use a probability distribution hence the initial value of 0.25.

If pages B, C, and D each only link to A, they would each confer 0.25 PageRank to A. All PageRank PR( ) in this simplistic system would thus gather to A because all links would be pointing to A.

$$PR(A) = PR(B) + PR(C) + PR(D).$$

But then suppose page B also has a link to page C, and page D has links to all three pages. The value of the link-votes is divided among all the outbound links on a page. Thus, page B gives a vote worth 0.125 to page A and a vote worth 0.125 to page C. Only one third of D's PageRank is counted for A's PageRank (approximately 0.083).

$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}.$$

In other words, the PageRank conferred by an outbound link L( ) is equal to the document's own PageRank score divided by the normalized number of outbound links (it is assumed that links to specific URLs only count once per document).

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}.$$

In the general case, the PageRank value for any page u can be expressed as:

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)},$$

i.e. the PageRank value for a page u is dependent on the PageRank values for each page v out of the set Bu (this set contains all pages linking to page u), divided by the number L(v) of links from page v.

1.5.4. Damping Factor

The Page Rank theory holds that even an imaginary surfer who is randomly clicking on links will eventually stop clicking. The probability, at any step, that the person will continue is a damping factor d. Various studies have tested different damping factors, but it is generally assumed that the damping factor will be set around 0.85.

The damping factor is subtracted from 1 (and in some variations of the algorithm, the result is divided by the number of documents in the collection) and this term is then added to the product of the damping factor and the sum of the incoming PageRank scores.

That is,

$$PR(A) = 1 - d + d\left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \cdots\right)$$

or (N = the number of documents in collection)

$$PR(A) = \frac{1-d}{N} + d\left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \cdots\right).$$

So any page's PageRank is derived in large part from the PageRanks of other pages. The damping factor adjusts the derived value downward. The second formula above supports the original statement in Page and Brin's paper that "the sum of all PageRanks is one". Unfortunately, however, Page and Brin gave the first formula, which has led to some confusion.

The formula uses a model of a random surfer who gets bored after several clicks and switches to a random page. The PageRank value of a page reflects the chance that the random surfer will land on that page by clicking on a link. 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 pages.

If a page has no links to other pages, it becomes a sink and therefore terminates the random surfing process. However, the solution is quite simple. If the random surfer arrives at a sink page, it picks another URL at random and continues surfing again.

When calculating PageRank, pages with no outbound links are assumed to link out to all other pages in the collection. Their PageRank scores are therefore divided evenly among all other pages. In other words, to be fair with pages that are not sinks, these random transitions are added to all nodes in the Web, with a residual probability of usually d = 0.85, estimated from the frequency that an average surfer uses his or her browser's bookmark feature.

So, the equation is as follows:

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

where p1,p2,...,pN are the pages under consideration, M(pi) is the set of pages that link to pi, L(pj) is the number of outbound links on page pj, 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

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

where R is the solution of the equation

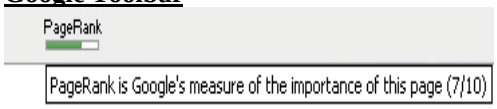
$$\mathbf{R} = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1,p_1) & \ell(p_1,p_2) & \cdots & \ell(p_1,p_N) \\ \ell(p_2,p_1) & \ddots & & \vdots \\ \vdots & & \ell(p_i,p_j) & \\ \ell(p_N,p_1) & \cdots & & \ell(p_N,p_N) \end{bmatrix} \mathbf{R}$$

where the adjacency function $\ell(p_i, p_j)$ is 0 if page pj does not link to pi, and normalized such that, for each j

$$\sum_{i=1}^{N} \ell(p_i, p_j) = 1,$$

i.e. the elements of each column sum up to 1.
This is a variant of the eigenvector centrality measure used commonly in network analysis. The values of the PageRank eigenvector are fast to approximate (only a few iterations are needed) and in practice it gives good results.
As a result of Markov theory, it can be shown that the PageRank of a page is the probability of being at that page after lots of clicks. This happens to equal t − 1 where t is the expectation of the number of clicks (or random jumps) required to get from the page back to itself.
The main disadvantage is that it favors older pages, because a new page, even a very good one, will not have many links unless it is part of an existing site (a site being a densely connected set of pages, such as Wikipedia). The Google Directory (itself a derivative of the Open Directory Project) allows users to see results sorted by PageRank within categories. The Google Directory is the only service offered by Google where PageRank directly determines display order. In Google's other search services (such as its primary Web search) PageRank is used to weight the relevance scores of pages shown in search results.
Several strategies have been proposed to accelerate the computation of PageRank.
Various strategies to manipulate PageRank have been employed in concerted efforts to improve search results rankings and monetize advertising links. These strategies have severely impacted the reliability of the PageRank concept, which seeks to determine which documents are actually highly valued by the Web community.

Google Toolbar



An example of the PageRank indicator as found on the Google toolbar.
The Google Toolbar's PageRank feature displays a visited page's PageRank as a whole number between 0 and 10. The most popular websites have a PageRank of 10. The least have a PageRank of 0. Google has not disclosed the precise method for determining a Toolbar PageRank value. Google representative Matt Cutts has publicly indicated that the Toolbar PageRank values are republished about once every three months, indicating that the Toolbar PageRank values are historical rather than real-time values.
The Google Directory PageRank is an 8-unit measurement. These values can be viewed in the Google Directory. Unlike the Google Toolbar, which shows the PageRank value by a mouseover of the green bar, the Google Directory does not show the PageRank as a numeric value but only as a green bar.

1.5.5. How is PageRank calculated?

To calculate the PageRank for a page, all of its inbound links are taken into account. These are links from within the site and links from outside the site.
PR (A) = (1-d) + d(PR(t1)/C(t1) + ... + PR(tn)/C(tn)) .
That's the equation that calculates a page's PageRank. It's the original one that was published when PageRank was being developed, and it is probable that Google uses a variation of it but they aren't telling us what it is. It doesn't matter though, as this equation is good enough.
In the equation 't1 - tn' are pages linking to page A, 'C' is the number of outbound links that a page has and 'd' is a damping factor, usually set to 0.85.

We can think of it in a simpler way: -
a page's PageRank = 0.15 + 0.85 * (a "share" of the PageRank of every page that links to it)
"share" = the linking page's PageRank divided by the number of outbound links on the page.

1.6. Trust Rank Algorithm

TrustRank is a link analysis technique described in a paper by Stanford University and Yahoo! researchers for semi-automatically separating useful webpages from spam.

Many Web spam pages are created only with the intention of misleading search engines. These pages, chiefly created for commercial reasons, use various techniques to achieve higher-than-deserved rankings on the search engines' result pages. While human experts can easily identify spam, it is too expensive to manually evaluate a large number of pages.

One popular method for improving rankings is to increase artificially the perceived importance of a document through complex linking schemes.

Google's PageRank and similar methods for determining the relative importance of Web documents have been subjected to manipulation.

TrustRank method calls for selecting a small set of seed pages to be evaluated by an expert. Once the reputable seed pages are manually identified, a crawl extending outward from the seed set seeks out similarly reliable and trustworthy pages. TrustRank's reliability diminishes as documents become further removed from the seed set. The researchers who proposed the TrustRank methodology have continued to refine their work by evaluating related topics, such as measuring spam mass.

Trust Rank is a major factor that now replaces PageRank as the flagship of parameter groups in the Google algorithm. (Note that a similar system is being used by Yahoo! Search as well). It is of key importance for calculating ranking positions and the crawling frequency of web sites.

1.6.1. Algorithm

Trust Rank is a semi-automatic algorithm used to separate reputable websites from spam. The main goal of TrustRank is based on the concept of trust attenuation. It starts by selecting a set of pages known as 'seed' pages whose spam status needs to be determined. A human expert analyses them first and tells the Algorithm if they are good pages or spam. Lastly, the algorithm looks for other pages that are similar to the good seed pages.
It is assumed that the further away one is from good seed pages, the less certain that a page is good. The following diagram represents this:

There are two pages (page 2 and 4) that are at most 2 links away from the good seed pages. As both of them are good, the probability that we reach a good page in at most 2 steps is 1. Similarly, the number of pages reachable from the good seed in at most 3 steps is 3. Only two of these (pages 2 and 4) are good, while page 5 is bad. Thus, the probability of finding a good page drops to 2/3. Therefore, trust is reduced as one

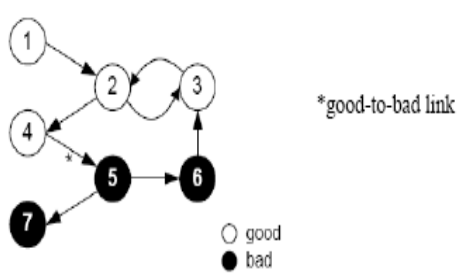


Figure 3: A web of good (white) and bad (black) nodes

moves further away from good seed pages.
The Trust Rank algorithm shown below computes trust scores for a web graph.

```
function TrustRank
input
        T       transition matrix
        N       number of pages
        L       limit of oracle invocations
        α_B     decay factor for biased PageRank
        M_B     number of biased PageRank iterations
output
        t*      TrustRank scores
begin
        // evaluate seed-desirability of pages
(1)     s = SelectSeed(...)
        // generate corresponding ordering
(2)     σ = Rank({1,...,N}, s)
        // select good seeds
(3)     d = 0_N
        for i = 1 to L do
                if O(σ(i)) == 1 then
                        d(σ(i)) = 1
        // normalize static score distribution vector
(4)     d = d/|d|
        // compute TrustRank scores
(5)     t* = d
        for i = 1 to M_B do
                t* = α_B · T · t* + (1 − α_B) · d
        return t*
end
```

It's inputs are the transition matrix, the number of web pages, the oracle invocation-which is the notion of human checking a page for spam by a binary search *oracle function* O over all pages *p* E v

$$O(p) = \begin{cases} 0 & \text{if } p \text{ is bad,} \\ 1 & \text{if } p \text{ is good.} \end{cases}$$

and finally *L*, *_B* and *MB* are parameters that control the execution.
The Select Seed function at the start identifies desirable pages fort he seed set and returns a vector **s**. It finds pages that will be most useful in identifying additional good pages. E.g. the following vector for figure (good-to-bad-link figure).
s = [0.08, 0.13, 0.08, 0.10, 0.09, 0.06, 0.02].
Step 2 uses the Rank function to reorder the elements of s in decreasing order, as shown;

s = [2, 4, 5, 1, 3, 6, 7]. Page 2 has the most desirable seed page as it has the maximum vector score 0,13, followed by page 4 and so on.
Step 3 uses the oracle function (fig x) and the *L* most desirable seed pages. The entries of the static score distribution vector **d** that correspond to good seed pages are set to 1.
Step 4 normalizes vector **d** so that its entries sum up to 1, and the following static score distribution vector is the outcome.
d = [0, 1/2, 0, 1/2, 0, 0, 0].
Step 5 evaluates TrustRank scores using a biased Page Rank computation with **d** replacing the uniform distribution.
Assuming that _*B* = 0.85 and *MB* = 20, the algorithm computes the following result:
t*= [0, 0.18, 0.12, 0.15, 0.13, 0.05, 0.05].
The good seed pages (2 and 4) have no longer a score of 1 but still have the highest score.
Further, a practical experiment showed that only 10 or 20 manual reviews can already filter out 1000 pf spam pages from the Google result pages by their escalating bad TrustRank. On the whole, this algorithm has shown that it can effectively identify a significant number of non-spam pages. It can be used in search engines either separately to filter the index, or in combination with Page Rank to rank results.

A buddy of mine pointed me to a white paper by Zoltan Gyongyi, Hector Garcia-Molina, & Jan Pederson about a concept called Trust Rank. Human editors help search engines combat search engine spam, but reviewing all content is impractical. Trust Rank places a core vote of trust on a seed set of reviewed sites to help search engines identify pages that would be considered useful from pages that would be considered spam. This trust is attenuated to other sites through link from the seed sites. Trust Rank can be use to

- Automatically boost pages that have a high probability of being good, as well as demote the rankings of pages that have a high probability of being bad.
- Help search engines identify what pages should be good candidates for quality review.

Some common ideas that Trust Rank is based upon:

- Good pages rarely link to bad ones. Bad pages often link to good ones in an attempt to improve hub scores.
- The care with which people add links to a page is often inversely proportional to the number of links on the page.
- Trust score is attenuated as it passes from site to site.

To select seed sites they looked for sites which link to many other sites. DMOZ clones and other similar sites created many non-useful seed sites.

Sites which were not listed in any of the major directories were removed from the seed set, of the remaining sites only sites which were backed by government, educational, or corporate bodies were accepted as seed sites.

When deciding what sites to review it is mostly important to identify high PR spam sites since they will be more likely to show in the results and because it would be too expensive to closely monitor the tail.

Trust Rank can be bolted onto PageRank to significantly improve search relevancy.

Search Engine Optimization is evolving at a fast pace. Several new changes have been incorporated to make search engines more efficient in terms of presenting useful information to their users.

1.7. Hits Algorithm

The HITS algorithm stands for 'Hypertext Induced Topic Selection' and is used for rating and ranking websites based on the link information when identifying topic areas. Kleinberg's hypertext-induced topic selection (HITS) algorithm is a very popular and effective algorithm to rank documents based on the link information among a set of documents. The algorithm presumes that a good hub is a document that points to many others, and a good authority is a document that many documents point to. Hubs and authorities exhibit a *mutually reinforcing relationship*: a better hub points to many good authorities, and a better authority is pointed to by many good hubs. To run the algorithm, we need to collect a base set,

including a root set and its neighborhood, the in- and out-links of a document in the root set.
HITS calculate hub and authority scores per query for the focused subgraph of the web. A good authority must be pointed to by several good hubs while a good hub must point to several goods authorities.

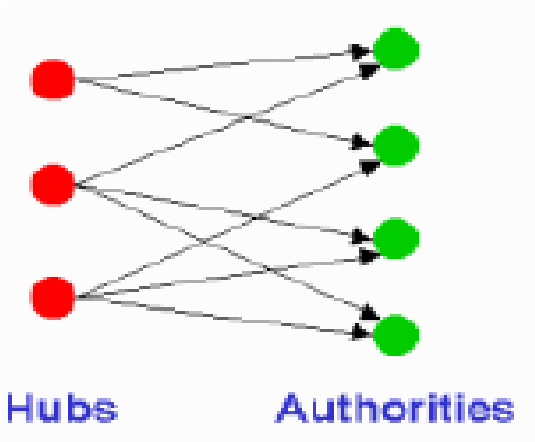


Figure 4: Hub and Authority pages

User queries are generally divided into two types. The specific query where the user requires exact matches and narrow information, and broad-topic query for user who look for narrow answers *and* information relation to the broad topic. HITS concentrates on the latter type and aims to find the most authoritative and informative pages for the topic of the query.

1.7.1Algorithm

HITS algorithm, can be stated as follows:
1. Using existing system, get the root set for the given query.
2. Add all the pages linking to and linked from pages in the root set, giving an extended root set or base set.
3. Run iterative eigenvector based computation over a matrix derived from the adjacency matrix of the base set.
4. Report the top establishment and hubs.

The first step in the HITS algorithm shows that the root set for a given query is taken from a search engine.
The second step basically expands the root set by one link neighborhood to form the base set.
The hub and authority value of page p is calculated in the following way:
Formula 1

$$A_p = \sum_{l \in Parp} H_l$$

where Ap stands for the authority value of page *p*,
Parp is the set of pages which point to *p* and are present in the base set,
and *l* for the number of links.

Formula 2

$$H_p = \sum_{l \in Chi_p} A_l$$

where Hp stands for the hub value of page *p*,
Chip the set of pages that *p* points to and which are present in the base
set and *l* for the number of links

Step 3 makes the n by n adjacency matrix E and its transposed matrix ET.
Let *Gq* be the graph corresponding to the base set for user query *q*.
Let *Va* represent the vector of authority values for all nodes in *Gq* and
Vh be the representation of hub values of all nodes of *Gq*. The following matrix computation vector can be calculated as follows:
Va = *ETVh* for formula 1
Vh = *Eva* for formula 2
Substituting values of *Va* and *Vh*,
Va = *ETEVa*
Vh = *EETVh*
Thus, *Va* will converge to principal eigenvector of *ETE* and *Vh* will converge to principal eigenvector of *EET* .
Finally, after determining elements with large values in the normalized eigenvector, the top authorities and hubs appear as outcomes.

In the HITS algorithm, the first step is to retrieve the set of results to the search query. The computation is performed only on this result set, not across all Web pages.

Authority and hub values are defined in terms of one another in a mutual recursion. An authority value is computed as the sum of the scaled hub values that point to that page. A hub value is the sum of the scaled authority values of the pages it points to. Some implementations also consider the relevance of the linked pages.

1.8. Comparison between Page Rank and Trust Rank

Page Rank (Google PR) and Trust Rank are the two main issues which are discussed frequently on various SEO forums. Although, both these

terms are related to SEO, they have different relationship with the actual rankings of your website. Your actual rankings are known as SERPs in the SEO world which stands for the Search Engine Ranking Positions of your website/webpages on Google/Yahoo. Let us first begin with Page Rank.

a) Page Rank – Page Rank is the numerical value, weighting 1-10 for any single website or web page. Page Rank represents the PR (Page Rank) strength of any given website. However, Page Rank should not be confused with the actual ranking strength of any website (i.e. your actual SERPs).

Page rank is generally calculated by search engines by analyzing and calculating various back links pointing towards a particular website. Google Page rank algorithm uses its own complex algorithmic formulas to calculate Page Rank of any given website or web page.

So far so good, but the real question comes up "Whether Page Rank has any relationship with the actual rankings of your website?" The answer is a big NO.

There are several hundred webmasters discussing increase/decrease in Page Rank on various forums, but none of them realize the importance of their actual SERPs as compared to Page Rank.

There are several hundred websites with low PR (1 or 2), and several of these websites genuinely overshadow other high-ranking competitor websites with higher Page Ranks (generally 4-5 or even higher than that).

There is a wrong misconception among several webmasters that higher the page rank, higher will be their rankings on search engines. The truth is that your increase or decrease in page rank has no relationship with the actual rankings of your website.

There are several webmasters who will argue that if any website sees a steep decline in their Page Rank, then it could be an indication of drop in their actual SERPs. If your website is having some major SEO issue related to content or links on your website, then your actual rankings (SERPs) will drop first. Once you observe a drop in SERPs, your Page Rank will eventually drop after a period of 2-3 months. Till the next update in Page Rank, the Google Toolbar will keep pulling PR Data from its stored database.

Your SERPs relies on the current status of your website whereas PR relies on the stored database which gets updated only once in every 2-3 or 3-4 months.

b) Trust Rank– The Google/Yahoo Trust Rank is one of the most important factors in determining actual rankings of your website i.e. your SERPs. Trust Rank is the Trust levels granted to any single website on the net. This Trust Rank helps certain websites to score higher rankings on search engines.

Let us try to anticipate some of the hidden factors which could influence Trust Rank of your website.

i) Age/History of your domain – The age of your domain is an important factor which can increase your Page Rank as well. There are several domains which are 5-6 years old and have higher rankings as compared to any other website within the same business category. Search Engines also rely on the history of your domain to detect any SPAM activity.

You can change everything in SEO, but you cannot change Age/History of any domain. Therefore search engines could rely on this factor to grant Trust status to certain websites.

ii) Quality of your back links – Now quality of your back links doesn't necessarily mean higher Page Rank links. In most of the cases it is observed that a website with Page Rank 2 is far much beneficial than getting a back link from a similar Page Rank 4 website. Search engines can easily detect any unnatural pattern in the back links of your website. If you purchase 50 back links with PR 3-4, then surely Google can detect it easily. The worst part is that your Page Rank is not related with Trust Rank, but these activities will lower your Page Rank as well as your Trust Rank. This paid link activity will increase your Page Rank, but it will lower you Trust Rank which ultimately decides ranking of your website.

iii) Content on your website – There are several article websites which have observed steep decline in their traffic on both Google and Yahoo. The reason behind this decline is the duplicate articles, which are submitted to several hundred websites by various webmasters and SEO. Several of these article websites have suffered consequences of duplicate penalty. Although there are several SEO's who will argue that duplicate penalty is just a myth. There are some who will say that duplicate penalty does exist, but it is more like having your duplicate content in supplement index. However, the truth is that if your content goes into supplement index, then obviously it will lower your Trust Rank.

One should concentrate on producing high quality content, rather than having bulk of content. If your content is good and informative, other webmasters will automatically start linking to your website.

iv) Optimization History of your domain – It could be possible that search engines store the optimization history of any single domain to determine their Trust Rank. If your website purchased 50,000 back links in 2006, then surely Google will keep an account of that activity to determine your Trust Rank.

1.9. Comparison between Page Rank and Hits Algorithm

HITS, like Page and Brin's PageRank, is an iterative algorithm based on the linkage of the documents on the web. However it does have some major differences:

- It is executed at query time, not at indexing time, with the associated hit on performance that accompanies query-time processing. Thus, the *hub* and *authority* scores assigned to a page are query-specific.
- It is not commonly used by search engines.
- It computes two scores per document, hub and authority, as opposed to a single score.
- It is processed on a small subset of 'relevant' documents, not all documents as was the case with PageRank.

Link graph features such as in-degree and PageRank have been shown to significantly improve the performance of text retrieval algorithms on the web. The HITS algorithm is also believed to be of interest for web search; to some degree, one may expect HITS to be more informative that other link-based features because it is query-dependent: it tries to measure the interest of pages with respect to a given query. However, it remains unclear today whether there are practical benefits of HITS over other link graph measures. This is even truer when we consider that modern retrieval algorithms used on the web use a document representation that incorporates the document's anchor text, *i.e.* the text of incoming links. This, at least to some degree, takes the link graph into account, in a query-dependent manner.

Comparing HITS to PageRank or in-degree empirically is no easy task.
There are two main difficulties: scale and relevance. Scale is important because link-based features are known to improve in quality as the document graph grows. If we carry out a small experiment, our conclusions won't carry over to large graphs such as the web. However, computing HITS efficiently on a graph the size of a realistic web crawl is extraordinarily difficult. Relevance is also crucial because we cannot measure the performance of a feature in the absence of human judgments: what is crucial is ranking at the top of the ten or so documents that a user will peruse. To our knowledge, this paper is the first attempt to evaluate HITS at a large scale and compare it to other link-based features with respect to human evaluated judgment. Our results confirm many of the intuitions we have about link-based features and their relationship to text retrieval methods exploiting anchor text. This is reassuring: in the absence of a theoretical model capable of tying these measures with relevance, the only way to validate our intuitions is to carry out realistic experiments. However, we were quite surprised to find that HITS, a query-dependent feature, is about as effective as web page in-degree, the most simpleminded query-independent link-based feature. This continues to be true when the

link-based features are combined with a text retrieval algorithm exploiting anchor text.

1.10. Conclusion

The ranking function has many parameters like the type-weights and the type-prox-weights. Figuring out the right values for these parameters is something of a black art. In order to do this, we have a user feedback mechanism in the search engine. A trusted user may optionally evaluate all of the results that are returned. This feedback is saved. Then when we modify the ranking function, we can see the impact of this change on all previous searches which were ranked. Although far from perfect, this gives us some idea of how a change in the ranking function affects the search results.
For search-engine optimization purposes, some companies offer to sell high Page Rank links to webmasters. As links from higher-PR pages are believed to be more valuable, they tend to be more expensive. It can be an effective and viable marketing strategy to buy link advertisements on content pages of quality and relevant sites to drive traffic and increase a Webmaster's link popularity. However, Google has publicly warned webmasters that if they are or were discovered to be selling links for the purpose of conferring Page Rank and reputation, their links will be devalued (ignored in the calculation of other pages' Page Ranks). The practice of buying and selling links is intensely debated across the Webmaster's community. Google advises webmasters to use the no follow HTML attribute value on sponsored links. According to Matt Cutts, Google is concerned about webmasters who try to game the system, and thereby reduce the quality and relevancy of Google search results.

Trust Rank is one of the popular method for improving rankings is to increase artificially the perceived importance of a document through complex linking schemes. Google's PageRank and similar methods for determining the relative importance of Web documents have been subjected to manipulation. Trust Rank can be bolted onto Page Rank to significantly improve search relevancy.

Additionally, in computing the level of relevance, we require a match between the query and the text on the expert page which qualifies the hyperlink being considered. This ensures that hyperlinks being considered are on the query topic. For further accuracy, we require that at least 2 non-affiliated experts point to the returned page with relevant qualifying text describing their linkage. The result of the steps described above is to generate a listing of pages that are highly relevant to the user's query and of high quality

.

1.11. References

1. Best of the Web 1994 -- Navigators http://botw.org/1994/awards/navigators.html
2. Bill Clinton Joke of the Day: April 14, 1997. http://www.io.com/~cjburke/clinton/970414.html
3. Bzip2 Homepage http://www.muraroa.demon.co.uk/
4. Google Search Engine http://google.stanford.edu/
5. Harvest http://harvest.transarc.com/
6. Mauldin, Michael L. Lycos Design Choices in an Internet Search Service, IEEE Expert Interview http://www.computer.org/pubs/expert/1997/trends/x1008/mauldin.htm
7. The Effect of Cellular Phone Use Upon Driver Attention http://www.webfirst.com/aaa/text/cell/cell0toc.htm
8. Search Engine Watch http://www.searchenginewatch.com/
9. RFC 1950 (zlib) ftp://ftp.uu.net/graphics/png/documents/zlib/zdoc-index.html
10. Robots Exclusion Protocol: http://info.webcrawler.com/mak/projects/robots/exclusion.html
11. Web Growth Summary: http://www.mit.edu/people/mkgray/net/web-growth-summary.html
12. Yahoo! http://www.yahoo.com/
13. [Abiteboul 97] Serge Abiteboul and Victor Vianu, *Queries and Computation on the Web*. Proceedings of the International Conference on Database Theory. Delphi, Greece 1997.
14. [Bagdikian 97] Ben H. Bagdikian. *The Media Monopoly*. 5th Edition. Publisher: Beacon, ISBN: 0807061557