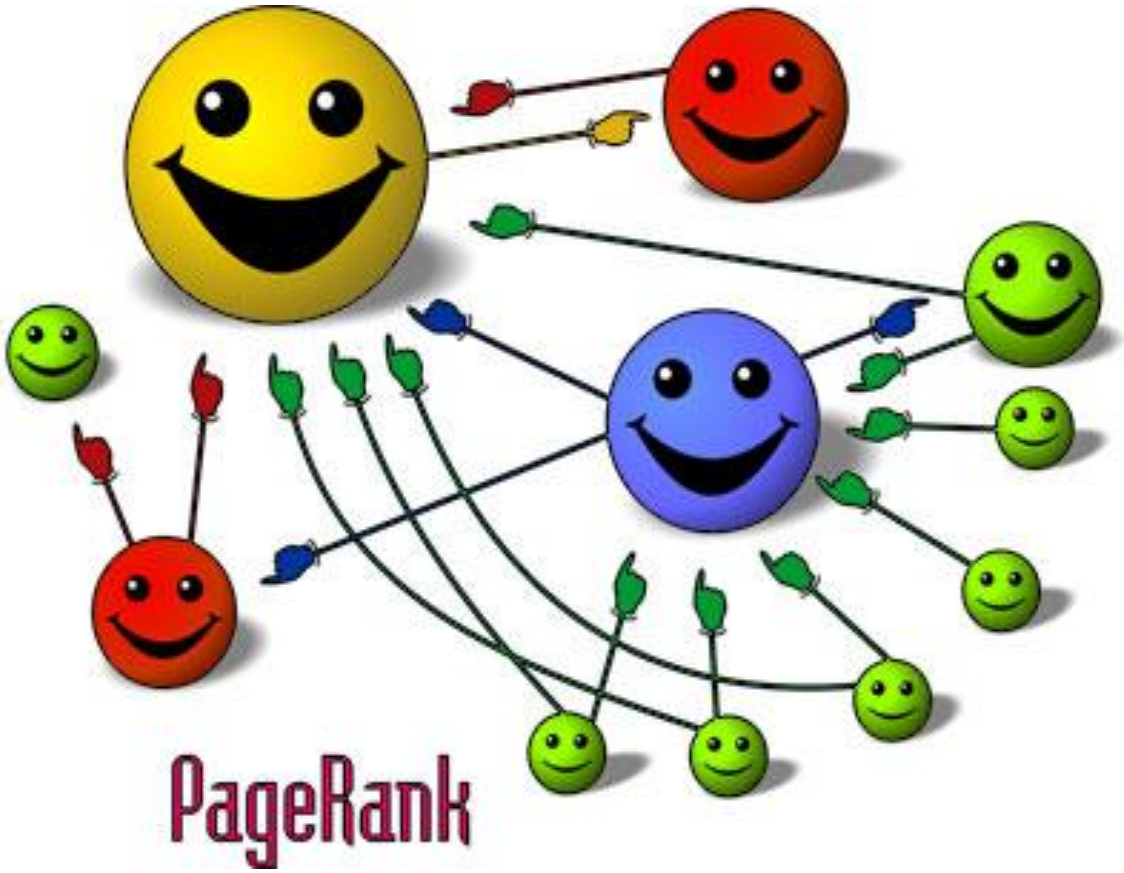
Study of Page Rank Algorithms

Objective:

The objective of this deliverable was to study the Google's and Yioop's Page Rank algorithm and suggest a method to rank the short links in Yioop.

Google's Page Rank



Topics Covered

- Background
- Introduction to Page Rank
- Algorithm
- Type of links
- Mathematics of Google Page Rank
- Internal Linking
- Additional Factors
- Yioop's Ranking method, my work and suggestion
- References

Background

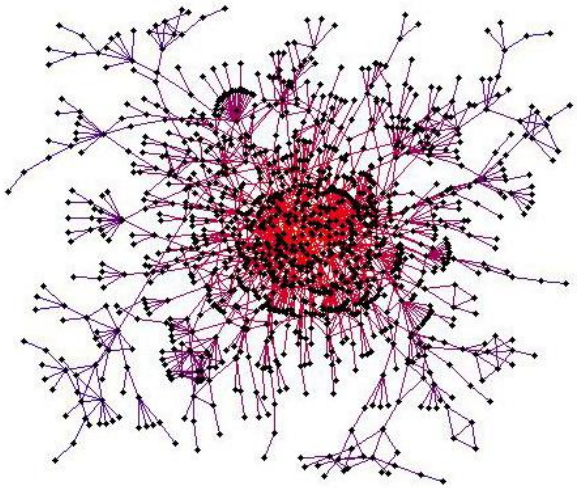
- Two popular algorithms were introduced in 1998 to rank web pages by popularity and provide better search results. They are:

  - HITS (Hypertext Induced Topic Search)
  - Page Rank

- HITS was proposed by Jon Kleinberg who was a young scientist at IBM in Silicon Valley and now a professor at Cornell University.

- Page Rank was proposed by Sergey Brin and Larry Page, students at Stanford University and the founders of Google.

- The Web's hyperlink structure forms a massive directed graph.[1]



- The nodes in the graph represent web pages and the directed arcs or links represent the hyperlinks.[1]

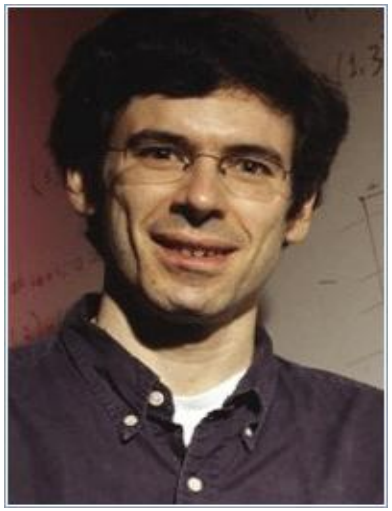
- Hyperlinks into a page are called inlinks and point into nodes and outlinks point out from nodes. They are discussed in details later. [1]

The theses underlying both HITS and Page Rank can be briefly stated as follows:

Page Rank

- Proposed by Sergey Brin and Larry Page
- Thesis: A web page is important if it is pointed to by other important web pages.[1]

HITS



- Proposed by Jon Kleinberg
- Thesis: A page is a good hub (and therefore deserves a high hub score) if it points to good authorities, and a page is a good authority if it is pointed to by good hubs.[1]

Introduction to Page Rank

- Page Rank is a numeric value that represents the importance of a page present on the web.

- When one page links to another page, it is effectively casting a vote for the other page.

- More votes implies more importance.

- Importance of the page that is casting the vote determines the importance of the vote.

- A web page is important if it is pointed to by other important web pages.
- Google calculates a page's importance from the votes cast for it.
- Importance of each vote is taken into account when a page's Page Rank is calculated.
- Page Rank is Google's way of deciding a page's importance.
- It matters because it is one of the factors that determines a page's ranking in the search results.
- Page Rank Notation- “PR”

Algorithm

The original Page Rank algorithm which was described by Larry Page and Sergey Brin is given by

$$PR(A) = (1-d) + d\,(PR(T1)/C(T1) + \ldots + PR(Tn)/C(Tn))$$

where,

- PR(A) – Page Rank of page A
- PR(Ti) – Page Rank of pages Ti which link to page A
- C(Ti) - number of outbound links on page Ti
- d - damping factor which can be set between 0 and 1

A simple way of representing the formula is, (d=0.85)

*Page Rank (PR) = 0.15 + 0.85 * (a share of the Page Rank of every page that links to it)*

✓The amount of Page Rank that a page has to vote will be its own value * 0.85.

✓This value is shared equally among all the pages that it links to.

✓Page with PR4 and 5 outbound links > Page with PR8 and 100 outbound links.

✓The calculations do not work if they are performed just once.

✓Accurate values are obtained through many iterations.

✓Suppose we have 2 pages, A and B, which link to each other, and neither have any other links of any kind.

✓Page Rank of A depends on Page Rank value of B and Page Rank of B depends on Page Rank value of A.

✓We can't work out A's Page Rank until we know B's Page Rank, and we can't work out B's Page Rank until we know A's Page Rank.

✓But performing more iterations can bring the values to such a stage where the Page Rank values do not change.

✓Therefore more iterations are necessary while calculating Page Ranks.

Types of Links

(1) Inbound links or Inlinks

- Inbound links are links into the site from the outside.
- Inlinks are one way to increase a site's total Page Rank.
- Sites are not penalized for inlinks.

(2) Outbound links or Outlinks

- Outbound links are links from a page to other pages in a site or other sites.

(3) Dangling links

- Dangling links are simply links that point to any page with no outgoing links.

Mathematics of Google Page Rank

It includes the following topics:

- Original Summation Formula for Page Rank
- Matrix Representation of the Summation Equations
- Problems with the Iterative Process
- Notation of Page Rank Model
- Adjustments to the Basic Model
- Computation of Page Rank Vector

The page rank equation is as follows,

$$\boldsymbol{\pi}^{T} = \boldsymbol{\pi}^{T}(\alpha \boldsymbol{S} + (1-\alpha)\boldsymbol{E})$$

Original Summation Formula for Page Rank

The Page Rank of a page P_i , denoted $r(P_i)$ is the sum of page ranks of all pages pointing into P_i ,

$$r(P_i) = \sum_{P_j \in B_{P_i}} \frac{r(P_j)}{|P_j|}$$

where,

B_{P_i} - the set of pages pointing to P_i

$|P_j|$ - the number of outlinks from page P_j

$r(P_j)$ value is unknown in the beginning of the calculation

- All pages are given equal page rank 1/n.
- n is number of pages in Google's index of Web.
- The equation in the previous slide is used to compute $r(P_i)$ for each P_i in the index.
- The equation is iteratively applied substituting the previous values.
- The following notation can be used to define the iterative procedure.

$$r_{k+1}(P_i) = \sum_{P_j \in B_{P_i}} \frac{r_k(P_j)}{|P_j|}$$

where,

$r_{k+1}(P_i)$ - Page rank of P_i at iteration k+1 and with initial ranks 1/n

Example

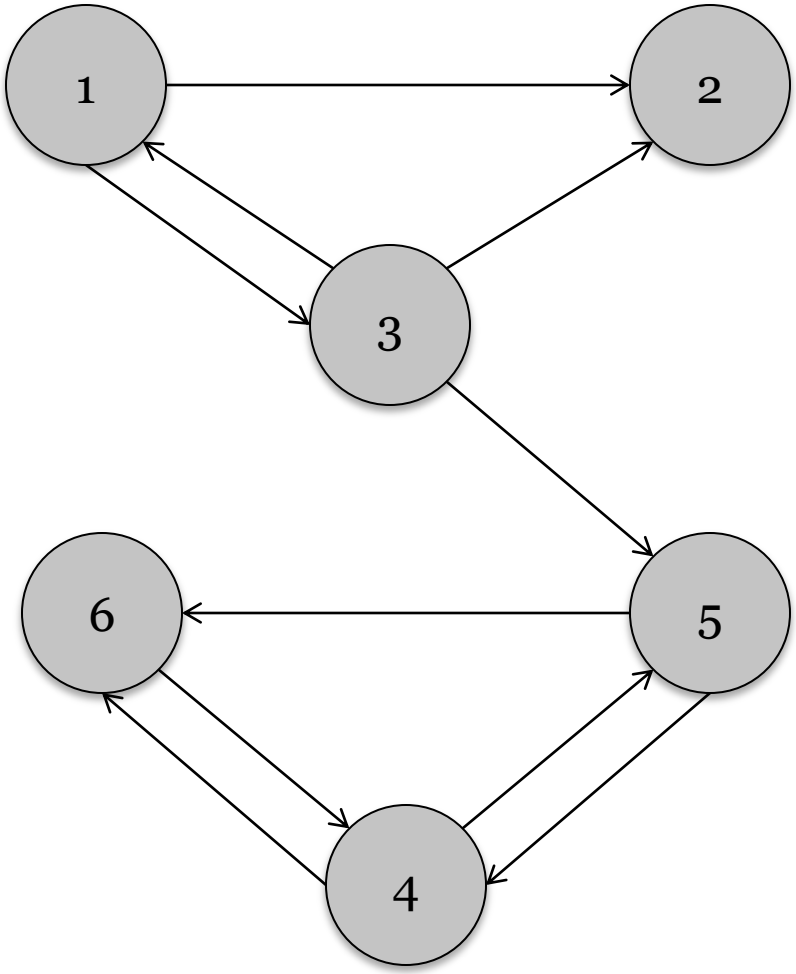


Fig1: Directed graph representing web of six pages

Iteration 0	Iteration1	Iteration2	Rank at iteration 2
R0(P1)=1/6	R1(P1)=1/18	R2(P1)=1/36	5
R0(P2)=1/6	R1(P2)=5/36	R2(P2)=1/18	4
R0(P3)=1/6	R1(P3)=1/12	R2(P3)=1/36	5
R0(P4)=1/6	R1(P4)=1/4	R2(P4)=17/72	1
R0(P5)=1/6	R1(P5)=5/36	R2(P5)=11/72	3
R0(P6)=1/6	R1(P6)=1/6	R2(P6)=14/72	2

Table 1: First few iterations using the iteration formula on the graph with six pages

Matrix Representation of the Summation Equations

- The summation symbol $\sum$ is replaced with matrices.
- At each iteration, a Page Rank vector (single 1xn vector) holding all the page rank values is computed.
- A hyperlink nxn matrix H and a 1xn row vector π^T are introduced.
- H is a row normalized hyperlink matrix with $H_{ij} = {}^{1}/_{|P_i|}$, if there is a link between node I to node j and 0 otherwise.

➢The H matrix for the graph in Figure 1 is given by,

H =

	P1	P2	P3	P4	P5	P6
P1	0	1/2	1/2	0	0	0
P2	0	0	0	0	0	0
P3	1/3	1/3	0	0	1/3	0
P4	0	0	0	0	1/2	1/2
P5	0	0	0	1/2	0	1/2
P6	0	0	0	1	0	0

Using the matrix notation the iteration equation can be re-written as,

$$\pi^{(k+1)T} = \pi^{(k)T} H$$

Few observations obtained from the Matrix equation are as follows,

- Each iteration involves one vector matrix multiplication, which requires $O(n^2)$ computation and n is the size of square matrix H.

- H is a very sparse matrix including many 0s. It requires minimal storage and matrix multiplications involving sparse matrices requires O(nnz(H)) computation, where nnz(H) is the number of non zeros in H.

- The iterative process is a simple linear stationary process. It is the classical power method applied to H.

- H looks like a stochastic transition probability matrix for a Markov chain. The dangling nodes in a network create the zero rows in the matrix.

Problems with the Iterative process

Two problems occurred when Brin and Page started the iterative process with $\pi^{(0)T} = 1/ne^T$, where e^T is the row vector of all 1s. They are:

- Rank Sink
- Cycles

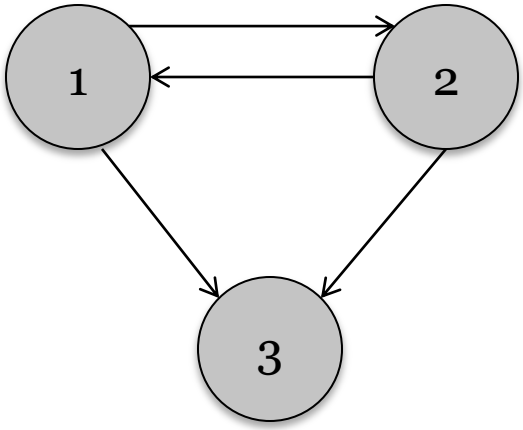


Fig 2: Rank Sink

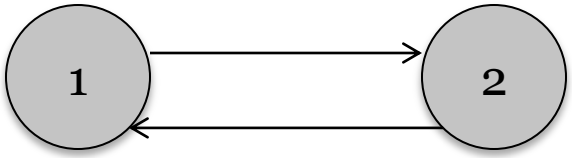


Fig 3: Cycles

Rank Sink

- One page accumulates more page rank at each iteration monopolizing the score.

- In Figure 2, the dangling node 3 is a rank sink.

Cycles

- In figure 3, nodes 1 and 2 form an infinite loop or cycle.

- The iteration process with these nodes will not converge irrespective of how long the process is run.

Early adjustments were made to the model to solve these problems.

Notation for the Page Rank Problem

H	Very sparse, raw sub stochastic hyperlink matrix
S	Sparse, stochastic, most likely reducible matrix
G	Completely dense, stochastic, primitive matrix called the Google matrix
E	Completely dense, rank-one teleportation matrix
n	Number of pages in the engine's index= order of H, S, G, E
α	Scaling parameter between 0 and 1
π^T	Stationary row vector of G called the Page Rank vector
a^T	Binary dangling node vector

Adjustments to the Basic Model

Random Surfer Model

- A random surfer is one who bounces along randomly following the hyperlink structure of the Web.

- The random surfer chooses one of the several outlinks present in a page.

- The importance of the page is determined by the proportion of time spent by the surfer on that page.

- Brin and Page used this notion of random surfer to describe the adjustments made to their basic model.

There are two problems with the random surfer model:

- A random surfer is caught when he encounters a dangling node such as an image, pdf, data tables etc.

- A random surfer completely abandons the hyperlink method and moves to a new browser and enter the URL in the URL line of the browser (teleportation).

Two adjustments were made to the basic page rank model to solve these problems.

- Stochasticity adjustment : Solves the dangling links problem

- Primitivity adjustment : Solves the teleportation problem

Stochasticity Adjustment

- In this adjustment, the 0^T rows of matrix H are replaced with $1/n\ e^T$ making H stochastic.

- This adjustment now allows the random surfer to hyperlink to any page at random after entering a dangling node.

- For the previous example of a web consisting of six nodes the stochastic matrix S is given by,

S =

	P1	P2	P3	P4	P5	P6
P1	0	1/2	1/2	0	0	0
P2	1/6	1/6	1/6	1/6	1/6	1/6
P3	1/3	1/3	0	0	1/3	0
P4	0	0	0	0	1/2	1/2
P5	0	0	0	1/2	0	1/2
P6	0	0	0	1	0	0

Mathematically, the stochastic matrix S is created from a rank one update to H. S is given as,

$$S = H + a(1/ne^T)$$

where a_i =1, if page i is a dangling node and,

a_i =0, otherwise

- S is a combination of the original hyperlink matrix H and a rank-one matrix $1/ne^T$

- S matrix alone cannot guarantee the convergence results.

- For this reason, another adjustment called primitivity adjustment has been done to the page rank model.

Primitivity Adjustment

- In this adjustment, Brin and Page introduced a new matrix G, called the Google matrix.

$$G=\alpha S+(1-\alpha)1/n ee^T$$

where α is a scalar between 0 and 1.

$\alpha=0.6 \Rightarrow$ random surfer follows the hyperlink structure of the Web 60% of the time and teleports to a random new page 40% of the time.

- The teleportation is random and the teleportation matrix $E= 1/n ee^T$
- The Google matrix G is stochastic, irreducible, aperiodic, primitive, completely dense and artificial.

Therefore, Google's adjusted Page Rank method is,

$$\pi^{(k+1)T} = \pi^{(k)T}G \text{ (power method applied to G)}$$

Applying this method to the example in the previous slides with $\alpha=0.9$, primitive matrix G is calculated as,

$G =$

1/60	7/15	7/15	1/60	1/60	1/60
1/6	1/6	1/6	1/6	1/6	1/6
19/60	19/60	1/60	1/60	19/60	1/60
1/60	1/60	1/60	1/60	7/15	7/15
1/60	1/60	1/60	7/15	1/60	7/15
1/60	1/60	1/60	11/12	1/60	1/60

- The Page Rank vector is given by,

$$\pi^T = (0.3721 \quad 0.5396 \quad 0.04151 \quad 0.3751 \quad 0.206 \quad 0.2862)$$

- Therefore, the pages in the example can be ranked as,

$$(4 \quad 6 \quad 5 \quad 2 \quad 3 \quad 1)$$

- The computation of page rank involves repeatedly applying Google's normalized variant of the web adjacency matrix to an initial guess of the page ranks.

This summarizes Google's Page Rank method.

Computation of the Page Rank vector

Finally, the Page Rank problem can be stated in two ways,

1. Solve the following eigenvector problem for π^T.

$$\pi^T = \pi^T G,$$
$$\pi^T e = 1$$

2. Solve the following the linear homogenous system for π^T.

$$\pi^T (1-G) = 0^T,$$
$$\pi^T e = 1$$

- In the first system,
  - Goal: Find the normalized dominant left-hand eigenvector of G corresponding to the dominant eigenvalue $\lambda_1 = 1$.
- In the second system,
  - Goal: Find the normalized left-hand null vector I-G.
- Both the systems are subject to the normalization equation $\pi^T e = 1$
- The power method which is one of the oldest and simplest iterative methods for finding the dominant eigenvalue and eigenvector of a matrix is used for the computation of the Page Rank Vector.

Internal Linking

✓ A website has a maximum amount of Page Rank that is distributed between its pages by internal links.

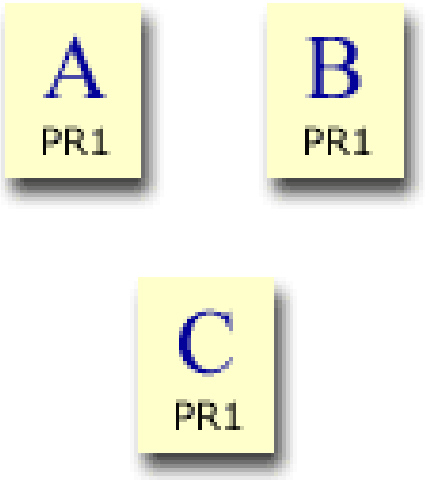
✓The maximum amount of Page Rank in a site increases as the number of pages in the site increases.

✓By linking poorly, it is possible to fail to reach the site's maximum Page Rank, but it is not possible to exceed it.

✓Few examples can illustrate the Page Rank concept and also the importance of internal linking of pages.

Example 1

✓If each of them have a rank of 1 then the site's total rank is 3.



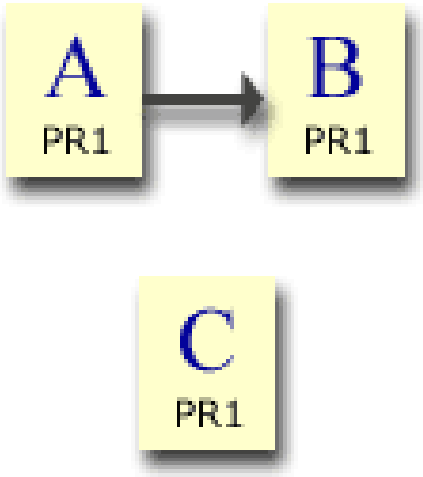
✓But if d=0.85 we see that each of them get a rank of just 0.15 .

✓This is because of the absence of internal linking between pages in the site

✓So the total PR for the site will be 0.45 instead of 3, which represents wastage of potential Page Rank.

Example 2

✓If each of them have a rank of 1 then the site's total rank is 3.



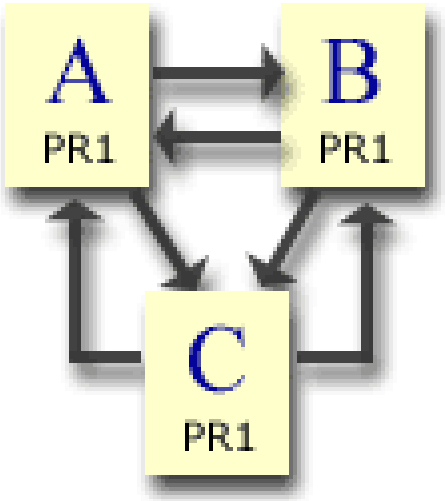
✓If d=0.85 we get the following PRs in the first iteration:

- PR(A)=0.15
- PR(B)=1
- PR(C)=0.15

✓After 100 iterations the PR(A) and PR(C) would be the same but PR(B) would be 0.2775

✓So the total PR for the site will be 0.5775 which is better than the previous one but still very less when compared to 3.

Example 3



✓If each of them have a rank of 1 then the site's total rank is 3.

✓If d=0.85 we get the following PRs in the first iteration:

- PR(A)=1
- PR(B)=1
- PR(C)=1

✓After any number of iterations the PRs of all the pages would remain same and also achieved the maximum a page can have.

✓It shows that good internal linking in a site would improve the page rank.

Yioop's Ranking method, my work and suggestion

- Yioop creates a word index and document ranking as it crawls and does not consider ranking as a separate step.

- Yioop groups all the links and documents associated with a URL into one group.

- The score computed is the sum of all the scores of individual documents.

- Previously, when a short link is encountered by Yioop, its URL was crawled and a raw URL was displayed in the search results as explained in deliverable 1.

- This assigned the rank to the short link instead of the original link.

- After creating a patch for Yioop, the original link associated with the short link is assigned to the URL to be crawled.

- This helps it assign the rank to the original link.

- This works in the case of bit.ly links and few other short links but encounters problems with few websites when the original link always redirects to another link.

- This problem in Yioop needs to be handled and the original link should be retrieved.

- After retrieving the original link, a higher weight can be assigned to the original link than the other links (shortened links, redirected links, etc) .

Additional Factors

Some of the additional factors which can influence Page Rank are:

- ✓ Visibility of a link
- ✓ Position of a link within a document
- ✓ Importance of a linking page
- ✓ Up-to-dateness of a linking page

References

(1) *Google's Page Rank and Beyond* by Amy N.Langville and Carl D.Meyer

(2) http://www.webworkshop.net/pagerank.html#how_is_pagerank_calculated

(3) http://en.wikipedia.org/wiki/PageRank

(4) http://pr.efactory.de/e-further-factors.shtml


Thank you