Ranking the Websites in Search Results using PageRank Algorithm

Vincent Endrahadi - 13515117
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13515117@std.stei.itb.ac.id ; vincentendrahadi01@gmail.com

***Abstract*—According to most of the websites' ranking website, search engine websites dominated the top 100 of most visited websites in the world. Search engine websites is so popular because it is the fastest way to find information. Other than fast, it is also easy to use. For example, if you want to know about the results of football matches, you can just type the information or anything related to that match and search engine will give you relevant websites. However, have you ever wonder that how is the hyperlink in the search results sorted? You have your website placed on page 20, but other websites can be placed on the first page? This paper will explain how the hyperlink was ranked using PageRank Algorithm.**

***Keywords*—PageRank , Search Engine, Link, ranking.**

I. Introduction

Internet had turned into a necessity for many people from all over the world. Internet is an interconnected network that connect many computers, World Wide Web (WWW), and many other things, including databases. Internet have grown all over the way from connecting 2 computers to a massive connection that connect people all over the world. Internet now can be used to buy things, banking, watch movies, a form of marketing and many other uses.

One of the most useful things in the internet is the search engine. Archie was the first created search engine in 1990 by Alan Emtage, student at McGill University in Montreal. Search engine have been improving up until now as the popularity never faded. Up until today there are more than 200++ search engines with many different purposes such as job search, shopping, questions & answers and many more. According to internetlivestats.com, internet search queries in 2012 reached 1.2 trillion queries with average of 1 second search time. This number shows how fast and popular search engines nowadays.

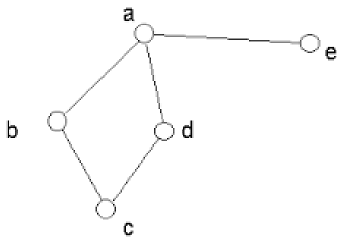
In searching queries, usually we will get many results whether it is a link, images, videos, books and many other kinds of things. It is usually classified with the same kinds such as, links will be shown with other links, images with other images and those search results can reach millions of results. The results is divided into pages and the most relevant will be in the few first pages. In order to sort that massive number of results, it needs a lot of computers and times, also a good algorithm such as PageRank Algorithm. This process is called Search Engine Optimization (SEO).

II. Basic Theories

A. Definition of Graph

Graph is a data structure that consists of nodes and edges where edges connects between one and another nodes. Graph is often used as a model of pairwise model.

In another way, G(V,E) is graph that consist of set of Vertex(nodes) that is not empty and set of Edges which can be empty.



Graph can be divided into several kinds based on its containment of loop and directed edges.

1. Based on graph containment of loop
   a. Simple Graph
      Graph that did not contain a loop edge. A loop edge is an edge that connect only a node.
   b. Un-simple Graph
      Graph that consists of edge that is a loop edge.

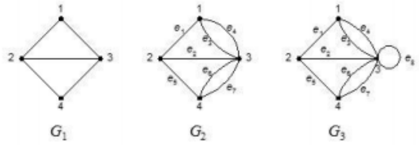


Fig 2.1 G_1 Simple Graph, G_2 and G_3 Un-simple graph

2. Based on directed edges.
   a. Directed Graph

Is a graph that it’s edges is directed from a node to a certain node.

b. Undirected Graph

Is a graph which the direction of the edges is ignored.

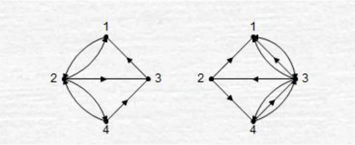


Fig 2.2 Directed Graph

B. Terminology in Graph

1. Adjacent

Two nodes are called adjacent if there is a edge that connect two of them.

2. Incidency

For every e, where e = (vj,vk), e is called incidence with vj/vk if e is the edge that connect vj with vk.

3. Isolated Vertex

Isolated Vertex is a vertex that did not have any edge that connect it with the other vertex.

4. Null Graph and Empty Graph

Empty Graph is a condition where the graph consists of several nodes but there are no edge that connect them. Null Graph is a graph that consist of no vertices and edges.

5. Degree

Degree of a Vertex is the sum of all edge that connect it with other vertex.

6. Path

Path with the length of n is the sum of edge that it go through from a vertex vj to vertex vk in which consists of all vertex in the graph.

7. Cycle/Circuit

Cycle is a path that start from vertex vj and will end up in vertex vj after go through all other nodes in the graph. The length of a circuit is the sum of edges that it go through.

8. Connected

Two vertex (vi, vj) is connected if there is a edge that connects them.

A graph (G) is connected if every pair of vertex (vi, vj) have a path that connect them else it is a disconnected graph.

9. Subgraph

Suppose that we have graph G(V,E) and G1(V1,E1). G1 is a subgraph of G if V1 is subset of V and E1 is subset of E.

10. Complement

Suppose we have a graph G2(V2,E2). Complement of graph G1 in point 9 is G2 where V2 = V-V1 and E2=E-E1.

11. Spanning Subgraph

Subgraph G1 is called spanning subgraph of G if G1 contain all the of vertex of G(V1 = V).

12. Cut-Set

Cut-set is set of edges that if was removed from the graph will result the graph to be disconnected.

13. Weighted Graph

Every edges of weighted graph is given a value. In this case, it can be used as a limit of how many times it can be go through or as a counter in a directed graph, how many vertices was directing to its place.

C. Special Graphs

1. Complete Graph

A simple graph that all of its vertex have edges that connect it to the remaining vertex. Complete graph with n number of vertex is symbolized as $K_{n.}$ Sum of K_n edges is n(n-1)/2.

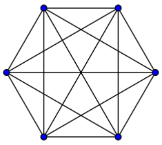


Fig 2.3 Complete Graph

2. Cycle Graph

A simple graph that degree of all of its vertex is 2. Cycle Graph is symbolized as $C_{n.}$

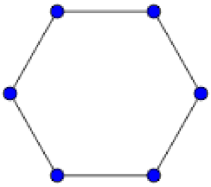


Fig 2.4 Cycle Graph

3. Regular Graph

A simple graph that degree of all its vertex is the same. If the degree of a vertex is r then the sum of edges in the graph is nr/2.

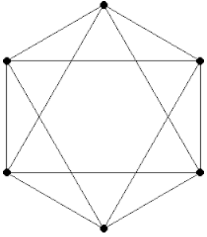


Fig 2.5 Regular Graph

4. Bipartite Graph

A graph that set of its vertices can be divided into two subsets of V1 and V2 and all the edges connect V1 and V2.

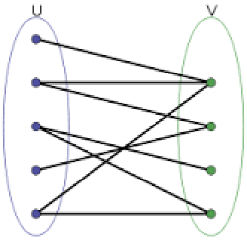


Fig 2.6 Bipartite Graph

III. PAGERANK ALGORITHM

1. PageRank Algorithm

PageRank Algorithm was described by Lawrence Page and Sergey Brin. PageRank Algorithm is also one of the first used in Google and the best known by people.

PageRank Algorithm used “vote” from many other website to determine the credentials and relevance of the content to rank the website. Suppose that we have a link from the page u to the page v. it can be viewed that v can be an important page that people needed. However, the importance of page u cannot be determined, therefore in order to determine the importance of every page require iterative fixed-point computation.

The PageRank Algorithm that is given in several publication is :

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

Where
PR(A) is the PageRank of Page A
PR(TI) is the PageRank of page T1 which has link to page A.
C(T1) is the outbound links in page T1.
d is a damping factor which can be set between 0 and 1.

From the algorithm above, we can see that the result of PR(A) will always be reduced as it is multiplied with d. The algorithm shows that PageRank Algorithm rank the page of a website rather than a whole website.

From PR(T1)/C(T1) we can see that, the more link that directed to page A will be beneficial to the PageRank of page A. However, if the linker website has many outbound links to the other page, it will decrease the PageRank of page A. This shows that the more websites that link to page A as its main link with no other links in that website the higher PageRank of page A.

2. Damping Factor

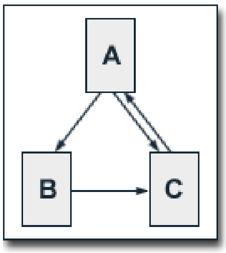
Lawrence page and Sergey Brin use a very intuitive explanation of PageRank Algorithm in their publication. Quoting from http://pr.efactory.de/e-pagerank-algorithm.shtml, the algorithm is explained this way :

> “Lawrence Page and Sergey Brin consider PageRank as a model of user-behavior, where the user can click at the random links with no regard toward the content.
>
> The random surfer visits a web page with a certain probability which derives from the page's PageRank. The probability that the random surfer clicks on one link is solely given by the number of links on that page. This is why one page's PageRank is not completely passed on to a page it links to, but is devided by the number of links on the page.
>
> So, the probability for the random surfer reaching one page is the sum of probabilities for the random surfer following links to this page. Now, this probability is reduced by the damping factor d. The justification within the Random Surfer Model, therefore, is that the surfer does not click on an infinite number of links, but gets bored sometimes and jumps to another page at random.
>
> The probability for the random surfer not stopping to click on links is given by the damping factor d, which is, depending on the degree of probability therefore, set between 0 and 1. The higher d is, the more likely will the random surfer keep clicking links. Since the surfer jumps to another page at random after he stopped clicking links, the probability therefore is implemented as a constant (1-d) into the algorithm. Regardless of inbound links, the probability for the random surfer jumping to a page is always (1-d), so a page has always a minimum PageRank.”

The damping factor is usually set to 0.85.

3. Characteristic of PageRank and its Computation

Suppose that we a 3 web pages, page A,B, and C. Page A links to Page B and C, Page B links to page C and Page C links to Page A. The damping factor is set to 0.5.

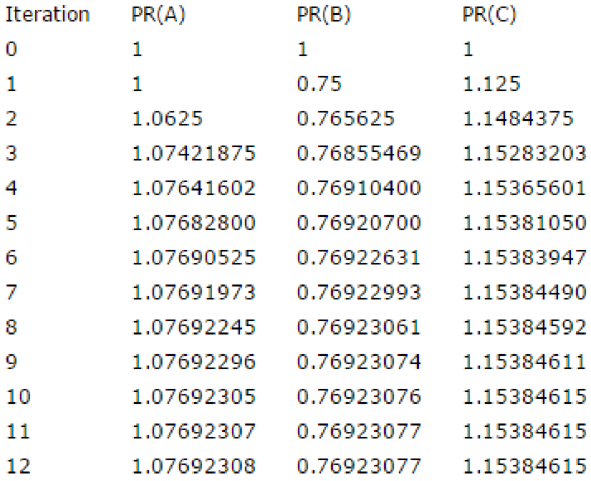


The equation of the calculation is :
PR(A) = 0.5 + 0.5 PR(C)
PR(B) = 0.5 + 0.5 (PR(A) / 2)
PR(C) = 0.5 + 0.5 (PR(A) / 2 + PR(B))

PR(A) = 14/13 = 1.07692308
PR(B) = 10/13 = 0.76923077
PR(C) = 15/13 = 1.15384615

From the result above, we can see that the sum of all PageRank is 3, which is the sum of all pages. The example above also can be solve intuitively, however, in practice, there are millions even billions of web which is impossible to determine the PageRank intuitively.

Google used approximative iterative computation as the size of the actual web. Each page is assigned an initial PageRank. The iterative process of example above can be seen in figure below, the initial PageRank is set as 1.



Iteration	PR(A)	PR(B)	PR(C)
0	1	1	1
1	1	0.75	1.125
2	1.0625	0.765625	1.1484375
3	1.07421875	0.76855469	1.15283203
4	1.07641602	0.76910400	1.15365601
5	1.07682800	0.76920700	1.15381050
6	1.07690525	0.76922631	1.15383947
7	1.07691973	0.76922993	1.15384490
8	1.07692245	0.76923061	1.15384592
9	1.07692296	0.76923074	1.15384611
10	1.07692305	0.76923076	1.15384615
11	1.07692307	0.76923077	1.15384615
12	1.07692308	0.76923077	1.15384615

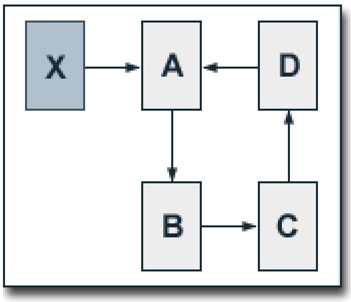
From the figure above, we can see that after 2 iteration we can get a good approximation of the PageRank. According to Lawrence Page and Sergey Brin, 100 or more iterative is needed in order to value the whole web.

Average of PageRank of webpage is 1. The minimum is (1-d). Therefore, the maximum PageRank is N+(1-d). This can be obtain if every website in the world links to this page and this page also link to itself.

4. The Inbound Links

From the algorithm, we can see that each additional inbound links always increase that page PageRank.

Suppose that we have a website consisting of Page A,B,C and D, linked to each other in circle. The initial PageRank of these page is 1. Then, we add a page X



which have a link to page A. Assume that PageRank of X is 10 and it is constant, with damping factor is 0.5.

PR(A) = 0.5 + 0.5 (PR(X) + PR(D))
= 5.5 + 0.5 PR(D)
PR(B) = 0.5 + 0.5 PR(A)
PR(C) = 0.5 + 0.5 PR(B)
PR(D) = 0.5 + 0.5 PR(C)

PR(A) = 19/3 = 6.33
PR(B) = 11/3 = 3.67
PR(C) = 7/3 = 2.33
PR(D) = 5/3 = 1.67

From the example above, we can see that PageRank of X which is 5 is passed on from A to B, and B to C which improve their PageRank. The boost of PageRank given by X is affected by the number of damping factor. If we change the damping factor in the example above to 7.5, it will boost the PageRank of A,B and C even more.

PR(A) = 0.25 + 0.75 (PR(X) + PR(D)) = 7.75 + 0.75 PR(D)
PR(B) = 0.25 + 0.75 PR(A)
PR(C) = 0.25 + 0.75 PR(B)
PR(D) = 0.25 + 0.75 PR(C)

PR(A) = 419/35 = 11.97
PR(B) = 323/35 = 9.23
PR(C) = 251/35 = 7.17
PR(D) = 197/35 = 5.63

With the damping factor of 0.5, the sum of all PageRank is given by

PR(A) + PR(B) + PR(C) + PR(D) = 14

The accumulated PageRank of all pages is increased by 10. With the damping factor of 0.75, the sum of all PageRank is given by

PR(A) + PR(B) + PR(C) + PR(D) = 34

The accumulated PageRank of all pages is increased by 30. From example given above, we know that the accumulated PageRank of all pages is increased by

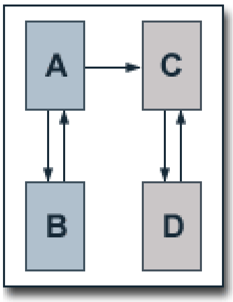
(d/(1-d)) * (PR(X)/C(X))

Where PR(X) is the PageRank of page additionally linking to one page of one side, and C(X) is the number of its outbound links. The formula only valid if and only the addition link point to a close system website. For example, bbc.com/news point to a blog that only link to itself.

IV. THE OUTBOUND LINKS

It is true that Inbound links of a page influence its PageRank, it also true that outbound links do have some impact.

Suppose that we have 2 websites, with each have two web pages. The first consists of page A and page B, the other consists of page C and page D. Both pages are link to each other and Page A link to page C. Assume that the damping factor is 0.75.



PR(A) = 0.25 + 0.75 PR(B)
PR(B) = 0.25 + 0.375 PR(A)
PR(C) = 0.25 + 0.75 PR(D) + 0.375 PR(A)
PR(D) = 0.25 + 0.75 PR(C)

PR(A) = 14/23
PR(B) = 11/23
Sum of the PageRank of the first site is 25/23

PR(C) = 35/23
PR(D) = 32/23
Sum of the PageRank of the second site is 67/23

The accumulated PageRank for both sites is 4. This shows that adding a link has no effect on the accumulated PageRank of the web. However, page C and D is benefited from the link while page A and B suffer loss of PageRank.

From the algorithm in inbound links, the PageRank benefit for a closed system given by addition links is

(d/(1-d)) * (PR(X)/C(X))

Where X is the linker page, PR(X) is the PageRank of page X and C(X) is the outbound links of page X. This value also represent the PageRank loss of page X because this page add a link to an external page.

V. Conclusion

It is clear that the amount of inbound links and outbound links affect the ranking of search results in search engine. By increasing the amount of inbound links in our webpage, we can climb into top search results in search engines. By being in the top search results give many opportunities to increase our web visitors. It is known by people that the most important and relevant information will be shown in top few pages. Hence, people rarely will go to the latter pages. Ranking up in search results can be used as marketing strategy of a company or even to promote an event.

VII. Acknowledgment

The author would like to express his gratitude to the God for giving the author the power to finish this paper, to his parents for giving him many support mentally and physically. The author also wanted to thanks Dr. Ir Rinaldi Munir as the lecturer of Discrete Mathematics who give me teachings and insight to complete this paper.

References

[1] Rogers, Ian, “The Google Pagerank Algorithm and How It Works”. Accessed at 6 December 2016. http://www.cs.princeton.edu/~chazelle/courses/BIB/pagerank.htm
[2] Sobek, Markus, The Pagerank Algorithm
Accessed at 8 December 2016.
Link : http://pr.efactory.de/e-pagerank-algorithm.shtml
[3] Sobek, Markus, The Effect of Inbound Links
Accessed at 8 December 2016.
Link : http://pr.efactory.de/e-inbound-links.shtml
[4] Sobek, Markus, The Effect of Outbound Algorithm
Accessed at 8 December 2016.
Link http://pr.efactory.de/e-outbound-links.shtml
[5] Rosen, Kenneth H., Global Edition Discrete Mathematics and Its Aplications, 7 th edition. New York : McGraw-Hill, 2013.
[6] Alhawarizmi, F, M(2014), “Analisis Graf Berarah pada Algoritma Pagerank di Mesin Pencari”
Accessed at 8 December 2016.
[7] Munir,Rinaldi , “Diktat Kuliah IF 2120 Matematika Diskrit”, Informatika Bandung : Bandung, 2007
[8] History of Seach Engines : From 1945 to Google Today
Link : www.searchenginehistory.com
Accessed at 8 December 2016.

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 8 Desember 2016

Vincent Endrahadi - 135515117