Pagerank Seo Algorithm : Issues, Complexity And Implementation

Mpemba Ngoma Luz[1], Kanyinda Kayembe Kam's[2], Likotelo Binene Camille[3], Nlandu Ngunda Jean[4], Nsumbu Lukamba Telesphore[5], Balanga Koko Joe[6], Mande Kumwimba Hydrice[7] Mayala Lemba Francis[8], Mbikayi Mpanya Jean *Marcel* [9], Engombe Wedi Boniface[10]

[1]Head of Works, Dept. of Informatics and Technology, Institut Supérieur Pédagogique(ISP)/ Mbanza-Ngungu/ DR Congo.
[2]Assistant2, Dept. of Informatics and Technology, , Institut Supérieur Pédagogique(ISP)/ Mbanza-Ngungu/ DR Congo.
[4,5]Assistant1, Dept. of Informatics and Technology, , Institut Supérieur Pédagogique(ISP)/ Mbanza-Ngungu/ DR Congo.
[7]Assistant1, Dept. of Mathematics and Computer Science, Université Pédagogique Nationale(UPN)/Kinshasa DR Congo.
[3,6]Assistant2, Dept. of Mathematics and Computer Science, Université Pédagogique Nationale(UPN)/Kinshasa DR Congo..
[8]Head of Works, Dept. Mathematics and Computer Science, Université Pédagogique Nationale(UPN)/Kinshasa DR Congo
[9]Professor, Dept. of Informatics, Institut Supérieur des Statistiques(ISS)/Kinshasa/ DR Congo. and
[10] Emeritus Professor, Dept. Mathematics and Computer Science,Université Pédagogique Nationale(UPN)/Kinshasa DR Congo.

Abstract

Today, the Internet users need an optimal search for information on the web. The display of web pages within a search engine is not a mystery. This implies good mathematical modeling and good knowledge of computer science for its implementation. The web is a directed graph that must be exploited. The matrices of its graph contain a structure of the links and the navigation of the Internet user. Considering the billions of hosted websites and the dynamism of the web, its links can be added at any time. Changing this link structure impacts the PageRank. Thus, for good stability, the algorithms must be improved. PageRank algorithm that displays a good web page Search Engine Optimization (SEO) taking into account the score of each page, pay attention to many researches that make improvements day by day. Even though it is a basic formula, the PageRank algorithm makes a successful business. In this paper, we had not only implemented this algorithm in python but also explain how it works and calculated its complexity.

Keywords: PageRank, Algorithm, Web, Search Engine, Graph, math, Python, Internet, SEO, score.

1. Introduction

Every day, many of us use web services for various reasons. For example, for a asked query, today's information systems compute hundreds of base score functions between that query and each document in a collection. Then combine these scores before assigning a final score to each of the documents [Massih at all, 2012].

The Information search, once reserved for specialists, has become one of the emancipated technologies of the 21st century. Each of us expects today to be able to find various information in record time on any type of subject through the Internet and these web services [Amani, 2001], [Beigdeder, 2012].

The Internet, being a global computer network, these web services operate in the form of a directed graph, which must be exploited through a search engine, for example [Bollobás, 1998],[Miller at al, 1997].

The Google search engine, one of the most powerful web services, offers Internet users the most relevant results based on a specific query. Several models are possible for web SEO, but this paper, talks about the PageRank scheduling algorithm.

This Google's algorithm is based on mathematical modeling, it intelligently sorts its results in order of relevance taking into account the best score of a web page j compared to the different links (href) of the outgoing pages i pointing to j [Amani, 2001], [Boarding 2022].Our contribution is twofold. First of all we will study the operation of the PageRank algorithm, its mathematical modeling as well as calculating its complexity. Then, normalize the web graph through an adjacent matrix and implement the algorithm in python language. And this, to calculate the PageRanks of each web page in order to find the best score for an optimal search for information.

2. General concepts

Definition 2.1 (Research models):are programs that help users find the information they are looking for in a collection of textual or multimedia documents. For a given information request, the purpose of these models is to return a subset of documents from the collection that could contain the information sought. The documents of this subset which actually contain the information sought are called relevant documents. [Maron and Kuhns, 1960], [Brini et al, 2006],[Salton and McGill 1986].

Definition 2.2 (crawler): According to the HTTPs protocol, a crawler retrieves Web pages that are stored in a list of URL addresses that the robot has beforehand. These pages are analyzed to extract the texts and links they contain. These extracted texts are then indexed and these links go through a series of tests to determine whether they should be added to the seed URL list or not. As pages are indexed, the corresponding URLs are removed from this list. The crawlers generally perform some type of normalization on URL addresses in order to avoid considering the same resource more than once [Amani, 2001],[Laudon and Laudon, 2013].

Definition 2.3 (Search engine) : Most search engines today use a two-step approach to order web pages with respect to a given user query. At the beginning of the algorithm, the content information of the pages is used to calculate scores of correspondence between these pages and the request. In the next step, new scores are then calculated, using the additional information of the links between pages. These scores, combined with previous scores, provide the final scores against which the pages are ordered [Massih, 2013]. [Bouzeghoub & Mosseri 2017].

PageRank algorithm Functioning : The Google's algorithm of search engine uses heavily the popularity index called "PageRank". PageRank [Brin and Page 1998] is the most popular information-based link ranking approach. Invented by Sergey Brin and Lawrence Rank, co-founder of Google company, it uses the hyperlink structure of the Web to build a Markov chain with a transition matrix P whose purpose is to give the probability that a user, randomly following the links of the Web pages on which it navigates, arrives on a particular page [Liu, 2011].

Initial formula: The initial formula for calculating PageRank was once given by Stanford University in a document entitled "The Anatomy of Large-Scale Hypertuel web Search Engine [Andrieu, 2010], [Martin and Chartier, 2016].

$$\mathrm{PR(A)} = (1 - \mathrm{d})/\mathrm{d} + \left(\frac{PR(T1)}{C(T1)} + \frac{PR(T2)}{C(T2)} + \cdots + \frac{PR(Tn)}{C(Tn)}\right)$$

Equation 1: Initial formula for calculating PageRank

Or :

- PR(A) equals the PageRank of page A;
- Tn (source pages) refers to the pages pointing (having set up a link) to page A (target page);
- C(Tn) represents the number of real links in the page Tn;
- d is a multiplying or damping factor. d is equivalent to Google's launch at 0.85. Google therefore justifies its formula: it can be evoked as representative of the behavior of an Internet user who would carry out a web browsing session and choose a web page, at random, then click on all the links it shows, and thus continue to click on all the links encountered. Eventually, this "crazy clicker" Internet user could get tired and start again, at one time or another, from a new starting page [Martin and Chartier, 2016].

In this myth, the probability that a page is visited by the Internet user is represented by its PageRank. And the "d" factor represents the fact that the crazy Internet user changes, at one time or another, the starting page to start again on a new surfing.

2.1. Definition and notation

The PageRank of a page j, denoted by PR(j) is the sum of the normalized PageRanks of all the pages pointing to this page. The normalized PageRank of a page is obtained by dividing its PageRank by the number of links leaving this page [Massih, 2017].

Thus, noting by: R(PR(1),….,PR(N))

The row vector whose component j$\epsilon\{1,\ldots,N\}$corresponds to the PageRank of page j, the recursive matrix calculation of the PageRanks is: R=RP

By initializing the vector R by the row vector where all the components are equal to $\frac{1}{N}$, for,

$R^{(0)} = \left(\frac{1}{N}, \ldots, \frac{1}{N}\right)$, the PageRanks can then be estimated iteratively: at each iteration l, the new vector$R^{(l+1)}$is simply calculated as$R^{(l+1)}$=$R^{(l)}$P. There is however no guarantee a priori that the preceding iterative calculation rule converges towards a unique solution. To guarantee this convergence, we rely on matricesergodic, which in practice is obtained by constructing P as below.

The construction of the matrix P takes place from the adjacency matrix A of the graph:A_{ij}= 1, ∃ a link from page i to page j, and 0 otherwise. For each page i containing outgoing links, the probability of borrowing a link leading to page j can be defined by:

International Journal for Multidisciplinary Research (IJFMR)

E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

$$\frac{A_{ij}}{\sum_{j=1}^{N} A_{ij}}.$$

Equation 2: The probability of consulting a link to j

The probability matrix P which accounts for this process is then a combination of these two types of passage:

$$P_{ij}=\begin{cases}\lambda\frac{A_{ij}}{\sum_{j=1}^{N} A_{ij}}+(1-\lambda)\frac{1}{N}\ si\ \sum_{j=1}^{n} A_{ij} \neq 0 \\ \frac{1}{N}\ Sinon\end{cases}$$

Equation 3: probability matrix pass combination

2.2. Algorithms and complexity

Definition 3.2.1 (Algorithmic): An algorithm is a succession of actions carried out in stages and which allows the realization of a task [Chen, 2022], [Cormen, 1994]. The algorithm below summarizes the iterative estimation of the PageRanks vector R. A proof of the convergence of the previous iterative rule with the matrix P defined as above can be found in [Langville and Meyer 2004]. λ (λ∈] 0.1[ ) is a damping factor usually set at 0.85.

Input :

- The adjacency matrix A of the directed web graph at a given instant;
- Damping factor λ; // *usually set to 0.85*
- Accuracy ϵ for the stopping criterion;

Initialization Calculate the probability matrix P by the equation1;

Vector PageRank R(0) = ,$\left(\frac{1}{N}, \dots, \frac{1}{N}\right)$I← 0.

```
Repeat
    R(l+1) = R(l)P
    l←l+1 Until || R(l+1) - Rl || <=
Exit : Vecteur PageRank R(l)
```

Algorithm 1: probability matrix calculation

Definition 3.2.2 (Complexity): The complexity of an algorithm is the number of elementary operations it must perform to complete a calculation based on the size of the input data. The complexity of an algorithm is the measure of the number of elementary operations it performs on the problem for which it was designed.

We can therefore now define more precisely the time complexity of an algorithm A. It is a function f, where f(n) is the maximum number of calculation steps (each instruction or elementary operation is associated with a cost called no computation) that A needs to solve a problem having an input of length n [Cormen at al, 1994],[Ngoie and Mpemba, 2015].

```
Bigin
d ← 0,85
oh ← G
ih ← G
N ← G
For i←0 to N
    opg[p] ← 1/N
    next i
    While interation > 0 do
        dp ← 0
        For i ← 0 to n(oh)
            dp ← dp + d * opg[p]/N
        next i
        For i ← 0 to N
                npg[p] ← dp + 1-d/N
        next i
            For i ← 0 to ih
                npg[p] ← npg[p] + d*opg[ip]/oh[ip]
            next i
        next i
        opg ← npg
        interation ← interation + 1
End
```

2.2.1. Algorithm of PageRank iteration calculations

Algorithm 2: PageRank interaction calculation

2.2.2. Calculation of complexity

f (n) = 1+1+1+1$\sum_{i=0}^{n}(1+1)$ + 1+ $\sum_{i=0}^{n}(1+1+1+1)$ + $\sum_{i=0}^{n}[(1+1+1+1)]$+ 1+1+1

= $4+\sum_{i=0}^{n} 2 + 1 + \sum_{i=0}^{n} 4 + \sum_{i=0}^{n}[4 + \sum_{i=0}^{n} 4] + 3$

= 8+ 2(n – 0 +1) + 4(n –0+1)+$\sum_{i=0}^{n} 4$ +$\sum_{i=0}^{n}[4(n-0+1)]$

= 8 + 2n + 2 + 4n + 4 + 4n +4 + $\sum_{i=0}^{n}(4n+4)$

= 18 +$10n + \sum_{i=0}^{n} 4n + \sum_{i=0}^{n} 4$

= 18 +10n+ 4n (n – 0 + 1) + 4 (n – 0 +1)

= 18 +10n+ 4n2 + 4n + 4n +4n

= 4n2 + 22n + 18

$\boldsymbol{o}(n^2)$: It is a quadratic complexity.

3. Modeling and implementation

Definition 4.1 (Graph): A graph is a triplet Γ = (V; E, N) where:

- V is the set of vertices of the graph; it will be convenient to use the notation V (Γ) to denote the set of vertices of the graph Γ;
- N is a set which is used to label the edges. N is the set of edges;
- E $\subset P_2(V)$ x N is the set of edges; notation E = E(Γ).

An edge a ∈ E is written a = ([x, y], n), x, y ∈V, n ∈ N ; x and y are the ends of a and n its label; a is incident to x and y; x and y are said to be adjacent; if x = y, the edge is a loop [Laforest, 2017].
On this, two edges a and b are said to be adjacent, if they have (at least) one identical end.

Definition 4.2 (Directed graph): A directed graph or digraph ($\vec{\Gamma}$ simply Γ) is a triple, $\vec{\Gamma}(V, \vec{E}, N)$ defined as follows:

- V is the set of vertices; notation V = V ($\vec{r}$);
- $\vec{E} \subset V \times V \times N$ is the set of arcs; notation (Γ);
- N is a set used to label arcs.

A bow a ∈ $\vec{E}$ will be noted a = ((x, y), n): the arc goes from x to y. In the following, we will often denote Γ for $\vec{r}$. Two incidence functions are used in this context:

$$i: \vec{E} \to V \quad et \quad t : \vec{E} \to V$$

Define a = ((x, y), n)) by:∀

- i(a) = x, the initial vertex of a,
- t(a) = y, the terminal vertex of a.

Similarly for undirected graphs, we say that arc a is incident to x and y and y is adjacent to x. Since a loop is an arc a such that i(a) = t(a). ∀ x, y ∈ V fixed, the set {a ∈ $\vec{E}$, i(a) = x, t(a) = y} of cardinality p is called p-arc; if p = 1, we speak of a simple arc; if p ≥ 2, it is a multi-arc.

Let x∈V. We define

d^-(x) = card{a∈−→ E , t(a) = x}, the indegree of x,

d^+(x) = card{a∈−→ E , i(a) = x}, the outdegree of x.

The degree of x is defined by d(x) := d−(x) + d+(x). If d(x) = 0, vertex x is said to be solitary.

If $d-$ (x) = 0 and d−(x) > 0, x is a sink.

If $d+$ (x) = 0 and d+(x) > 0, x is a source.

If it k ∃ ∈N such that vertex x, we have d + (x) = k, the digraph is said to be outgoing semi-regular. If We have the same property for incoming degrees, the digraph is incoming semiregular. The digraph is said to be constant if it is both incoming semi-regular and outgoing semi-regular [Gross and Yellen, 2007].

Scenario : In this article, we will develop the principle of an algorithm leading tothe estimation of these probabilities on a simple example consisting of a directed graph representing the structure of the hyperlinks of 8 Web pages from P1 to P8. The nodes of this graph illustrate the pages and the directed arcs represent the links between these pages. The Markov model then represents this graph as a square matrix P where an element π_{ij} is the transition probability from page i to page j.
The basic assumption used to construct the matrix P is that a user, being on any page of the web, clicks with equiprobability on the links leaving this page to arrive at another page.

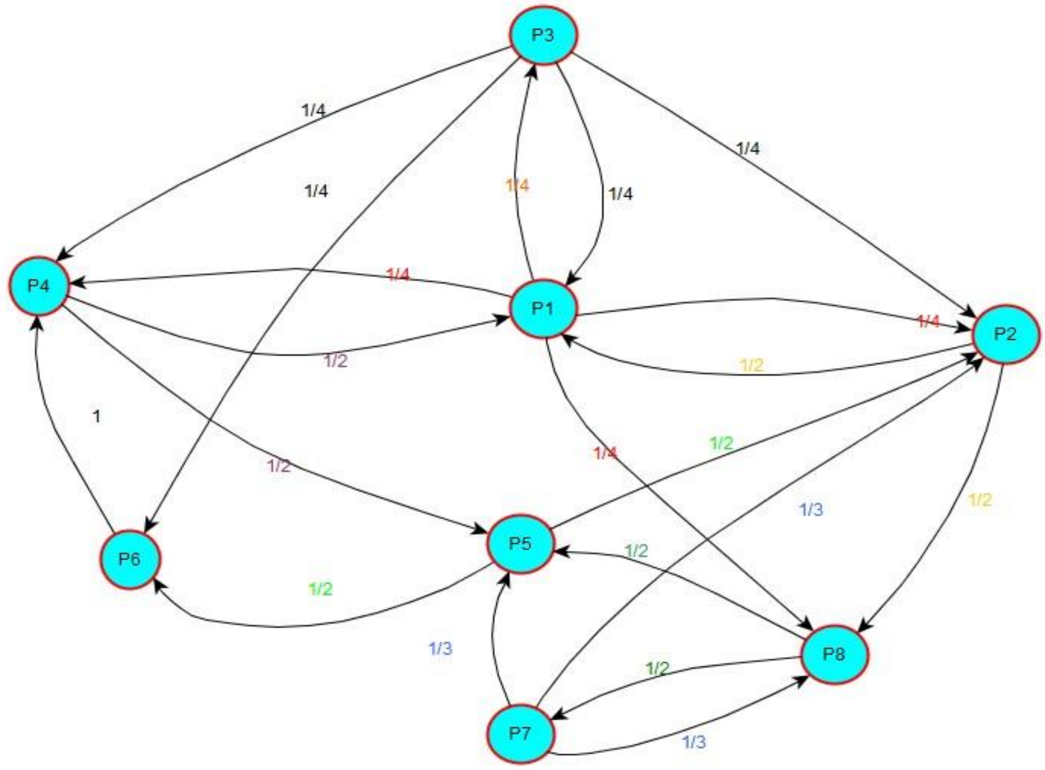

Figure 1: Oriented graph P normalized

3.1. Directed graph normalization

The normalized PageRank of a page is obtained by dividing its PageRank by the number of links leaving this page [Massih, 2013].

$$\mathbf{P} = \begin{pmatrix} 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{2} & 0 & \frac{1}{3} & 0 \\ \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{3} & \frac{1}{2} \\ 0 & 0 & \frac{1}{4} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \end{pmatrix}$$

Figure 2: adjacent matrix of normalized P graph

Let's calculate the pages of the directed web graph P taking into account the following elements:

PR(p)= $\sum_{i\in in(p)} \frac{PR(i)}{|out(i)|}$

Of which :

$\sum_{i\in in(p)}$ It is the sum of all elements which have a link with p denoted by ent(p)

$\frac{PR(i)}{|out(i)|}$ is the sequence for each element i, we take the PageRank of (i) divided by the number of outgoing links from page i

$$PR(p) = \frac{1-d}{n} + d \sum_{i\in ent(p)} \frac{PR(i)}{|sort(i)|}$$

$$PR(P1) = +0.85\times\frac{1-0,85}{8}\left(\frac{PR(P1)}{4} + \frac{PR(P3)}{4} + \frac{PR(P5)}{2} + \frac{PR(P7)}{2}\right)$$

PR(P2) = + 0.85 × $\frac{1-0,85}{8}\left(\frac{PR(P1)}{4}+\frac{PR(P3)}{4}+\frac{PR(P5)}{2}+\frac{PR(P7)}{2}\right)$

PR (P3) = +0.85 ×$\frac{1-0,85}{8}\left(\frac{PR(P1)}{4}\right)$

PR(P4) = +0.85×$\frac{1-0,85}{8}\left(\frac{PR(P1)}{4}+\frac{PR(P3)}{4}+\frac{PR(P6)}{1}\right)$

PR(P5) = +0.85×$\frac{1-0,85}{8}\left(\frac{PR(P4)}{2}+\frac{PR(P7)}{3}+\frac{PR(P8)}{2}\right)$

PR(P6) = +0.85×$\frac{1-0,85}{8}\left(\frac{PR(P3)}{4}+\frac{PR(P5)}{2}\right)$

PR(P7) = +0.85×$\frac{1-0,85}{8}\left(\frac{PR(P8)}{2}\right)$

PR(P8) = +0.85×$\frac{1-0,85}{8}\left(\frac{PR(P1)}{4}+\frac{PR(P2)}{2}+\frac{PR(P7)}{3}\right)$

3.2. Implementation

We had chosen the Python language in relation to its characteristics and its recent performances in mathematical modeling programming. Python is a language that to evolve [Swinnen, 2009].

Python source code at:https://github.com/jnn95/PageRank.git

3.2.1. Results (Calculation of nodes, edges and unit matrix of graph P)

The nodes of P are: ['P1', 'P2', 'P3', 'P4', 'P8', 'P6', 'P5', 'P7']

The edges of P are: [('P1', 'P2'), ('P2', 'P1'), ('P1', 'P3'), ('P3', 'P1'), ('P1', 'P4'), ('P4', 'P1'), ('P2', 'P8'), ('P3', 'P2'), ('P3', 'P6'), ('P6', 'P4'), ('P5', 'P4'), ('P8', 'P5'), ('P5', 'P2'), ('P7', 'P2'), ('P8', 'P7'), ('P7', 'P8'), ('P1', 'P8'), ('P5', 'P6')]

***There* matrix *unitary* is :** $\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

3.2.2. Result of Score Calculations (PageRank)

{'P1': 0.2252566341110866,
'P2': 0.1495245661586878,
'P3': 0.06661752682111585,
'P4': 0.1459826301138691,
'P5': 0.09039822672940236,
'P6': 0.06323766669962978,
'P7': 0.09039822672940236,
'P8': 0.16858452263680596}

*It is the **P1** page which has a better vote or a high score. It is therefore the page that gives a better web referencing taking into account the graph P.*

5. Conclusion

In this article, which deals with the issues, the complexity and the implementation of the PageRank algorithm, we showed how the Google research Engine calculates the PageRank. We also demonstrated that how its matrix is a combination of stochastic matrices of the structure of the links and the behavior of the Internet user in the web operation.

The complexity calculated in this paper, shows that it is quadratic given the algorithm has two nested loops. PageRank's formula is calculated iteratively as web pages keep growing, so a damping constant of 0.85 is used to lighten the load.

This convergence speed also gives the best ranking. The example of the web graph used in this paper shows that the web page P1 gives us a better SEO score of 0.2252566341110866, compared to the other pages. Note also that we have never finished calculating the PageRank of Web pages because:

- In terms of complexity, we do not know in advance the number of iterations, it depends on amarginerror;
- Then, it's a distributed algorithm, there is not a super Google computer which calculates all the PageRanks of all the web pages of a site, it is rather thousands of computers each of which calculates the subsets of the PageRanks and then put the results together; Once these scores are calculated, they can be sorted from largest to smallest and display good SEO to the Internet user.
- And finally, the web is dynamic (there are links that appear and disappear, there are pages that are deleted and others can be created).

PageRank algorithm remains a very open subject because the web has become our daily life and the search for the best information remains essential for everyone. Other improvements for the future may be a PageRank algorithm that can understand semantics by talking about the Semantic Web taking into account ontologies, in order to find accurate results in seconds instead of increasingly complex queries .

References

1. Andrieu O. Successful web referencing: SEO strategy and technique, Eyrolles, 2010.
2. Beigbeder, M. Software for IR, Fall School in Information Retrieval and Applications (EARIA), 2012.
3. Bollobás B., Modern Graph Theory. Springer, 1998.
4. Bordage F. , Ecoconception Web, 115 good practices, Eyrolles, Paris 2022.
5. Bouzeghoub M. & Mosseri R., Big data exposed, CNRS Editions, Paris, 2017
6. Brin, S. and Page, L. The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems, 33:107–117, 1998.
7. Brini, A., Boughanem, M. and Dubois, D. Possibilistic networks for an information retrieval model. In CORIA, pages 143–154, 2006.
8. Chen, Y.-Y., Gan, Q., & Suel, T. I/O Efficient Techniques for Computing Pagerank., 2002.
9. Cormen TH, Leiserson C., Rivest R.;Chretienne P.; Cazin X. Introduction àAlgorithmic Dunod, Paris, 1994.
10. Gross .JL and Yellen. J, Graph theory and its applications, second edition. CRC Press, 2007.
11. JL Gross & J. Yellen, Graph Theory and its Applications, second edition. CRC Press, 2007.
12. Langville, AN and Meyer, CD, Deeper inside pagerank. Internet Mathematics, 2004.
13. Laudon K v. and Laudon JP, essentials of management information systems, Pearson Education, 2013.
14. Maron, ME and Kuhns, J. L, On relevance, probabilistic indexing, and information retrieval. Journal of the ACM, 7(3) 216–244, 1960.

15. Martin AT, Chartier M., Web referencing techniques, Eyrolles, 2016
16. Massih R. Amini, Gaussier.E, Applications, models and algorithms - Data mining, decision-making and big data- Algorithms collection,Eyrolles, 2017.
17. Massih, Amini R., Gaussier É., Information retrieval Applications, models and algorithms, Eyrolles, Paris 2013.
18. Ngoie R B and Mpemba N.L, Complexity Study on Fibonacci's Sequence, (IJSIMR) Volume 3, Issue 10, October 2015, PP 5-13.
19. Salton, G. and McGill, MJ Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York, NY, USA, 1986.
20. Swinnen G., Learning to program with Python" from (third and fifth editions), formerly published by O'Reilly and now published by Eyrolles (ISBN 978-2-212-13434-6) ,2009.