Advanced Intelligent Search Method (AISM): A Method to Efficiently Search Authoritative Web Pages

Prachi Singhai[1], Dr. Gireesh Dixit[2]

[1]*M.Tech, Madhav Proudyogiki Mahavidyalaya, Bhopal, Madhya Pradesh, India*

[2]*Prof. And Head (Comp. Sc.), Madhav Proudyogiki Mahavidyalaya, Bhopal*

Abstract— Size of the web is huge and is growing very rapidly. With millions and billions of pages available on the web, it becomes very difficult for the users to use the rich hyper structure. Search engines like Google try their best to provide relevant information to the users according to the query posted by them, but in many cases search results are only satisfactory or even poor. Therefore, there is a need to find a more efficient method to retrieve the relevant information for the user. There are many algorithms available at present which are used by different search engines for link analysis like Page Rank (PR), Weighted Page Rank (WPR), Hyperlink-Induced Topic Search (HITS),Intelligence search method(ISM) etc. The objective of this research is to discover an efficient and better system to identify general web pages and to compare the results with existing algorithms. This new method is named as Advanced Intelligent Search Method (AISM).

Keywords—HITS, Page Rank, WPR, General Web Pages, Web Structure Mining, Link Analysis, ISM, AISM.

I. INTRODUCTION

1.1 Web Mining

We can define Web mining as mining of data present in the World Wide Web Database in the form of web pages and the data related to Web activity. Web data can be in the following forms:

1. Web pages content like text and images.
2. Intra page structure, which includes the HTML tags or XML tags.
3. Inter page structure, which is links from one page to another page.
4. Usage data, which describe access pattern of web page by the visitors on the Internet.

1.2 Web Content Mining

We can define Web Content Mining as the method of examining and investigating the content of Web pages. It may also include the results of web searching. The content of web pages may include text as well as graphics data. We can further divide Web Content Mining into two types, i.e. web page content mining and search results mining.

Web page content mining can be defined as conventional searching of web pages with the help of content. It can be used to improve the efficiency of search engines through various techniques. For example, the search engine may look into the <META> tag of web pages for the search keywords.

Search results mining can be defined as searching new web pages based on the results of a previous search.

We can use various data mining techniques to improve the results of Web Content Mining. It means it is always possible to improve efficiency, effectiveness and scalability of results.

1.3 Crawler

We can define Crawler as the program that traverses the hypertext structure of the web. Sometimes it also referred as a Spider or a Robot.

The Crawler starts from one page which is called seed page. It records all the links of the seed page and stores it in a queue. All the elements of the queue (which are different URLs of the links) are then taken one by one and the process is repeated recursively. This may result an infinite loop so we need some mechanism for stopping and returning from the recursive process.

1.4 Web Structure mining

We can define Web structure mining as mining information about the actual organization of pages on the Web. So Web structure mining is about creating a model of the Web organization.

We can use web structure mining for classification of web pages. It can also be used to measure similarity between documents.

There are various algorithms available for Web Structure Mining such as Page Rank, Weighted Page Rank, HITS etc.

II. SYSTEM MODEL

In this model indexing the web pages is done using an intelligent search strategy. This method first interprets the meaning of the search query and then index the web pages based on the interpretation. The new method can be integrated with any of the Page Ranking Algorithms to produce better and relevant search result and can work in any general database.

The method is tested by taking some sample queries. First the query is posted in original form to Google search engine and first thirty results are analyzed to find out total number of relevant pages.

III. PREVIOUS WORK

2.1 Page Rank Algorithm

This algorithm was developed by Sergey Brin and Lawrence Page at Stanford University and is named after Lawrence Page. Page Rank is a link analysis algorithm which extends the idea of citation analysis. It consider citation graph of the web as an important resource. In this algorithm a numerical rank is assigned to each element of set of documents which are linked together. So we measure relative importance of documents within a set.

The Page Rank algorithm is based on web graph which is created by considering web pages as vertices and hyperlinks as edges. The rank value of a page is indication importance of that page. The numerical weight of a page A is denoted by PGRN (A) or Page Rank of page A.

We can define Page Rank as follows:

Suppose page A has pages T1, T2, T3……., Tn that points to it, Then Page Rank of page A is given as follows:

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

Here

PGRN (Ti) – is page rank of page Ti
d - Damping factor, 0 <= d <= 1 (usually set to 0.85)
C (Ti) – is number of links going out of page (Ti)

DAMPING FACTOR

When a person is surfing the web he or she might click some links and at some point of time will eventually stop. The damping factor is the probability that the person will continue at any step. Different damping factors are tested by various studies but usually it is set to 0.85.

An Example:

Suppose we have 5 web pages A, B, C, D and E in a small system of hyperlink structure. Initially Page Rank of all the pages is same. So every page has initial rank of 1/5 i.e. 0.2.

Now suppose page A is pointed by pages B, C, D and E then these four links will transfer 0.2 page rank to A upon next iteration as follows:

PGRN (A)
=PGRN (B) + PGRN(C) + PGRN (D) + PGRN (E)
=0.2+0.2+0.2+0.2
=0.8

So Page Rank of page A will be 0.8 after first iteration.

Now suppose page B is pointing to A and C. So it will distribute its Page Rank equally (i.e. 0.1) to both these pages. Similarly if page E is pointing to all other four pages then equal distribution will be 0.05 to each of these pages. So in the next iteration:

PGRN (A) =PGRN (B)/2 + PGRN (C) + PGRN (D) + PGRN (E)/4

=0.1+0.2+0.2+0.05

=0.55

This process can go on recursively for calculating the Page Rank of other pages in the similar way. Therefore,

PGRN (A) = PGRN (B)/C (B) + PGRN (C)/C (C) + PGRN (D)/C (D)+ PGRN (E)/C (E)

Here C(x) is number of links going out of page x.

Based on the above example, the general formula for calculating Page Rank can also be given as follows:

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{C(v)}.$$

PR (u) is Page Rank of page u. Page Rank of page u is dependent on page rank values of page v that points to it. Here page v is an element of set of pages under consideration denoted by Bu.

It is important to note the recursive nature of this formula. It means for computing the Page Rank we need to know the page rank of other pages. Therefore, in our example we started with equal value of 0.5. In real situation we will need about 50 iterations to find out the final Page Rank.

Figure 1 show how these Page Rank calculations are happening.

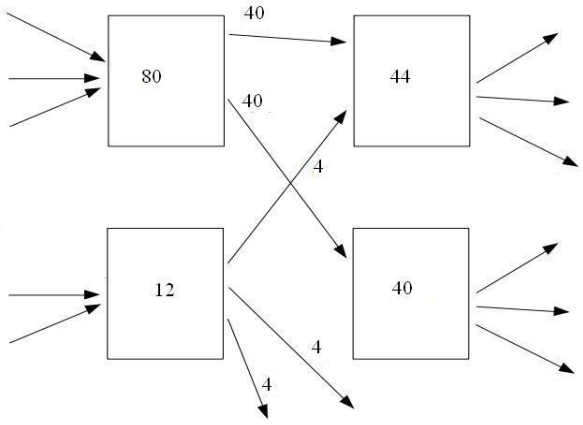


Fig. 1 : Simplified PageRank Calculations

The back-link coming from an important page is given higher weight age. Similarly the back links coming from non-important pages will be given less weight age.

The Page Rank forms a probability distribution all over the web pages so the sum of Page Ranks of all web pages will be one. The Page Rank of a page can be calculated without knowing the final value of Page Rank of other pages. It is an iterative algorithm which follows the principle of normalized link matrix of web. Page Rank of a page depends on the number of pages pointing to a page.

3.2 WEIGHTED PAGE RANK

This algorithm can be considered as an extension of Page Rank algorithm. It was proposed by Wenpu Xing and Ali Ghorbani. In this algorithm pages are assigned rank values on the basis of their importance as opposed to even division of ranks in Page Rank algorithm. As explained above, in Page Rank we start with giving all the pages equal ranks (0.2 in our example) so ranks are divided evenly. In WPR initial even distribution of ranks will not be there. Rather we start with giving higher rank value initially to an important page. So different page of the system will having be different weights. The weight of the page is decided in terms of incoming and outgoing links. This is denoted as W^i (m,n) and W^o (m,n) respectively.

W^i (m,n) – is the weight of entering link(m,n) . The calculation of W^i (m,n) is done on the basis of incoming links to page **n** and total number of incoming links to all the reference pages of page **m**. So we can apply following formula to calculate W^i (m,n):

$$W^{in}_{(m,n)} = \frac{I_n}{\sum_{p \in R(m)} I_p}$$

In the above formula I_n is total number of entering links of page n and I_p is total number of entering links of page p, the page p is an element of set of pages R(m), and R(m) is the reference page set of page m.

W^o (m,n) is defined similarly, which is the weight of exit link(m,n). So we can calculate W^o (m,n) on the basis of total number of exit links of page **n** and total number of exit links to all reference pages of Page **m**. so we can apply following formula to calculate W^o (m,n):

$$W^{out}_{(m,n)} = \frac{O_n}{\sum_{p \in R(m)} O_p}$$

In the above formula O_n is total number of ou links of page n, O_p is total number of outgoing links of page p, page p is an element of set of pages under consideration. The set of pages is denoted by R (m).

Finally WPR can be calculated as follows:

$$WPR(n) = (1-d) + d \sum_{m \in B(n)} WPR(m) W^{in}_{(m,n)} W^{out}_{(m,n)}$$

COMPARISON OF PAGERANK AND WEIGHTED PAGERANK

First of all we can classify set of pages under consideration into four types i.e. VR, R, WR and IR. The description of these types is given below:

1. VERY RELEVANT PAGES (VR): Some pages contain very important information can be classified as

VR (Very Relevant Pages). So these are most important pages in the set of pages under consideration.

2. RELEVANT PAGES I: Some pages are relevant but it is possible that they are not containing important information about the query posted by the user. So these pages are classified as R (Relevant Pages).

3. WEAKLY RELEVANT PAGES (WR): Some pages contain the keywords of the query posted by the user but they do not have any relevant information. Such pages are WR (Weakly Relevant Pages)

4. IRRELEVANT PAGES (IR): If page neither contain keywords of the query nor the relevant information then it is classified as IR (irrelevant Page).

The pages given by Page Rank and WPR algorithms are in sorting order according to their rank calculated on the basis of user query. So, it is very important that the user gets relevant pages and these relevant paged must be served in a proper sequence. Both the sequence and relevancy of pages is important.

We can compare Page Rank and WPR on the basis of relevancy rule, which is as follows:

RELEVANCY RULE: The Relevancy of a page is calculated on the basis of which class (i.e. VR, R, WR or IR) the page belongs to. The larger relevancy value will give better result. We can calculate the relevancy values as

$$k = \sum_{i \in R(p)} (n - i) * W_i$$

Here i is the i^{th} page in the result page-list Rl(p), n represents the first n pages chosen from the list Rl(p), and W_{ith} is the weight value of i^{th} page.

W_i = (v1, v2, v3, v4, v5)

Here, v1, v2, v3, v4 and v5 are the values assigned to a page if the page is VRl, Rl, WRl and IRl respectively. It is also obvious that

v1>v2
V2>v3
v3>v4

Experiments conducted on these algorithms prove that we get higher relevancy values with WPR as compared to Page Rank.

3.3 HITS (HYPER-LINK INDUCED TOPIC SEARCH)

This algorithm is also a link analysis algorithm proposed by Jon Klienberg. It classifies web pages into two categories called as hubs and authorities. Hubs are the pages that links to other important pages so they act as resource lists. Authorities are the pages containing important contents. A good hub page can be defined as a page which is having links pointing to many authoritative pages. Similarly, a good authority page is a page which is pointed by many good hub pages related to a given content. The concept of hubs as authorities can be understood with the help of Fig. 2 shown below. It may be possibly page may be a better hub as well as a good authority at the same time in others.

The HITS algorithm treats World Wide Web as directed a graph:

G= (V, E)

We can consider V is a set of vertices that indicant pages and E is set of edges that represent links.

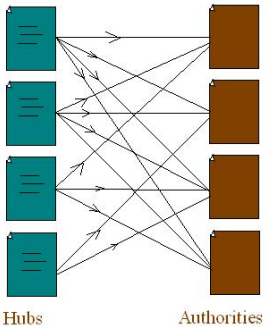


Fig. 2 : Hubs and Authorities

This algorithm has two steps:

1. Sampling Step: In this step we collect a set of relative pages for the given query.

2. Iterative Step: This step corresponds to finding out Hubs and Authorities pages. Following expressions are used to calculate the weight of Hub (H_p) and the weight of Authority (A_p).

$$H_p = \sum_{q \in I(p)} A_q$$

$$A_p = \sum_{q \in B(p)} H_q$$

here H_q is Hub Score of a page, A_q is authority score of a page, I(p) is set of reference pages of page p and B(p) is set of referrer pages of page p, the authority weight of a page is proportional to the sum of that link the hub weights of pages. Similarly a hub of a page is proportional to the sum of that link the authority weights of pages.

LIMITATIONS HITS ALGORITHM -Following are some of the limitations of HITS algorithm:

1. Hubs and authorities: It is difficult to differentiate between hubs and authorities because there are many pages in the internet which hubs as well as authorities.

2. Topic drift: There is a possibility that HITS may not produce the most relevant pages according to the query posted by the user because of equivalent weights.

3. Automatically generated links: Many times the links are generated automatically by server side programs and HITS gives them equal importance. It may not produce relevant results as per the query posted by the user.

4. Efficiency: In actual real time situation HITS algorithm is not very efficient.

COMPARISON

We can compare the algorithms discussed above on the following points:

1. Working of Algorithm: The Page Rank calculates scores at the time of indexing and results are stored according to importance of pages. The Weighted Page Rank also works in the same manner. The HITS computes Hub and Authority scores of n highly relevant pages on the fly.

2. Input Parameters: For Page Rank input parameter is Back Links Only. For Weighted Page Rank input parameter is Back Links and Forward Links. For HITS input parameter is Back Links, Forward Links and Content.

3. Complexity:
Page Rank – O (log N)
Weighted Page Rank - < O(log N)
HITS - < O(log N)

LIMITATIONS OF EXISTING METHODS

All the algorithms listed above may provide satisfactory performance in some cases but many times the user may not get the relevant information. The problem we all face when we search a topic in the web using a search engine like Google is that we are presented with millions of search results. First of all it not practically feasible to visit all these millions of web pages to find the required information. Second, when we visit few initial links shown in the search results, we may not get the relevant information.

Therefore, we all feel the requirement of a mechanism so that we can get the relevant information according to the query posted by us.

The major problem that we feel with all these algorithms is that none of them include that

"Intelligent Search Factor". By intelligent search we mean that there is a need for interpreting the inherent meaning of the query and indexing should be based on that.

IV. PROPOSED METHODOLOGY

The new method is named as "Advanced Intelligent Search Method (AISM)". In this method indexing the web pages is done using an intelligent search strategy. This method first interprets the meaning of the search query and then index the web pages based on the interpretation. The new method can be integrated with any of the Page Ranking Algorithms to produce better and relevant search result and can work in any general database.

The method is tested by taking some sample queries. First the query is posted in original form to Google search engine and first thirty results are analyzed to find out total number of relevant pages.

Then query is interpreted with the help of AISM database (4.2). This database is a simple database containing a table called Interpret_Query. The table has two columns: Original_Query and Interpretation. The getInterpretaion() method(4.3) takes Original query as parameter and it return interpretation of the query. The Java implementation of the method is given in section 4.3. The interpreted queries is posted again on the same search engine and first thirty results are analyzed again. The results are then compared to find out which method is better.

4.1 AISM ALGORITHM

Step 1: Input Search Query

Step 2: Generate interpretations of search queryion using getInterpretation() method and AISM Database given in the section 4.2 and 4.3.

Step 3: Post interpreted query to the search engine.

Step 4: generate the search results.

4.2 ISM Database

Table : Interpret_Query

Original_Query	Interpretation
Narayan Murthy except Infosys Founder	Narayan Murthy –Infosys
XML tutorial on website w3schools.com	XML tutorial site:w3schools.com
sites similar to facebook.com	related: www.facebook.com
Only PDF tutorial on Database Management System	"Database Management System" tutorial filetype:pdf
books on C#.Net from 2002 to 2010	C#.Net books 2002..2010
useful links on DBMS	inanchor:DBMS
Url including the word company	inurl: company
Indian classical dance form except bharatnatyam	Indian classical dance form – bharatnatyam

V. EXPERIMENTAL RESULTS

The experimental results are shown in the following table. The table has 4 columns. Description of each column is as follows:

1. Column 1(Original Query) : This column contain the original query posted by the user to the Google search engine.

2. Column 2(No. of relevant results in top 30 results): This column contains number of relevant results out of first 30 results given by the google search engine.

3. Column 3(Interpreted Query): This column contains the Interpreted query given by ISM which is posted again to the Google search engine.

4. Column 4(No. of relevant results in top 30 results): This column contain number of relevant results out of first 30 results given by the Google search engine after posting the interpreted query.

4.3 Java implementation of getInterpretation() Method.

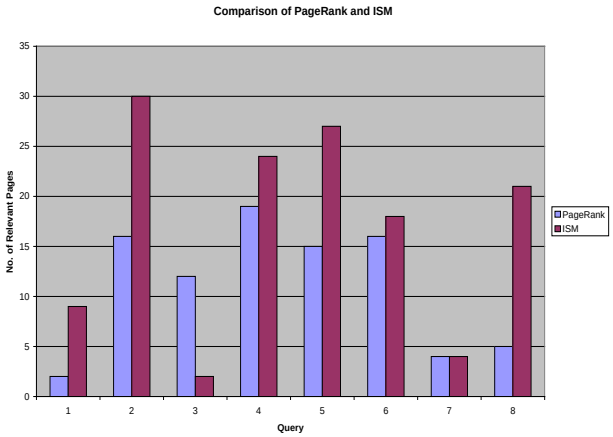
```
class AISM
{
 public String getInterpretation(String query)
 {
          try
          {
 Class.forName("sun.java.jdbc.odbc.JdbcOdbcDriver");
           Connection
c=DriverManager.getConnection("jdbc:odbc:ISMDSN","","
");
           Statement s=c.createStatement();
           ResultSet rs=c.executeQuery("Select * from
Interpret_Query where Original_Query=' " + query + " ' " );
          rs.next();
          String result=rs.getString(1);
          return result;
          }
          catch(Exception e)
          {
          System.out.println(e);
          }
          return null;
 }

}
```

TABLE: COMPARISON OF PAGERANK AND AISM

Page Rank		AISM	
Original Query	No. of relevant results in top 30 results(By PageRank)	Interpreted query (Using ISM)	No. of relevant results in top 30 results (By ISM)
Narayan Murthy except Infosys Founder	2	Narayan Murthy – Infosys	9
XML tutorial on website w3schools.com	16	XML tutorial site:w3schools.com	30
sites similar to facebook.com	12	related: www.facebook.com	2
Only PDF tutorial on Database Management System	19	"Database Management System" tutorial filetype:pdf	24
books on C#.Net from	15	C#.Net books	27

2002 to 2010		2002..2010	
useful links on DBMS	16	inanchor:DBMS	18
Url including the word company	4	inurl: company	4
indian classical dance form except bharatnatyam	5	Indian classical dance form – bharatnatyam	21
Total	89		135

The experimental results are also shown in following bar chart.



VI. CONCLUSION

It is clear from the above experiment that AISM produces better results in most of the cases and it fails only in few cases. This method can be implemented on the top of any existing searching algorithm to produce more relevant search results.

VII. FUTURE SCOPE

In future a proposal for implementing this method can be submitted to existing search engines so that it can be practically implemented by those search engines for better results.

REFERENCES

[1] Ashish Jain, Rajeev Sharma, Gireesh Dixit, Varsha Tomar, "Page Ranking Algorithms in Web Mining, Limitations of Existing methods and a New Method for Indexing Web Pages", International Conference on Communication Systems and Network Technologies, April 2013

[2] Shesh Narayan Mishra ,Alka Jaiswal and Asha Ambhaikar, "An Effective Algorithm for Web Mining Based on Topic Sensitive Link Analysis". International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 4, April 2012

[3] Tamanna Bhatia, "Link Analysis Algorithms For Web Mining". IJCST Vol. 2, Issue 2, June 2011.

[4] Rekha Jain, Dr. G. N. Purohit, "Page Ranking Algorithms for Web Mining", International Journal of Computer Applications (0975 – 8887),Volume 13– No.5, January 2011

[5] Pooja Sharma, Pawan Bhadana, "Weighted Page Content Rank For Ordering Web Search Result", International Journal of Engineering Science and Technology, Vol 2, 2010.

[6] Fabrizio Lamberti, Andrea Sanna and Claudio Demartini , "A Relation-Based Page Rank Algorithm for. Semantic Web Search Engines", In IEEE Transaction of KDE, Vol. 21, No. 1, Jan 2009.

[7] N. Duhan, A. K. Sharma and K. K. Bhatia, "Page Ranking Algorithms: A Survey, Proceedings of the IEEE International Conference on Advance Computing, 2009.

[8] Su Cheng,Pan YunTao,Yuan JunPeng,Guo Hong,Yu ZhengLu and Hu ZhiYu "PageRank, "HITS and Impact Factor for Journal Ranking", In proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering – Vol. 06, PP. 285-290, 2009.

[9] Lian-Wang Lee, Jung-Yi Jiang, ChunDer Wu, Shie-Jue Lee, "A Query-Dependent Ranking Approach for Search Engines", Second International Workshop on Computer Science and Engineering, Vol. 1, PP. 259-263, 2009. [26] NL Bhamidipati et al., "Comparing Scores Intended for Ranking", In IEEE Transactions on Knowledge and Data Engineering, 2009.

[10] E. Horowitz, S. Sahni and S. Rajasekaran, "Fundamentals of Computer Algorithms", Galgotia Publications Pvt. Ltd., pp. 112-118, 2008.

[11] H Jiang et al., "TIMERANK: A Method of Improving Ranking Scores by Visited Time", In proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, 12-15 July 2008.

[12] Dalibor Fiala, "Web Mining and Its Applications to Researchers Support", Technical Report No. DCSE/TR-2005-06, April 2005

[13] Miguel Gomes da Costa Júnior and Zhiguo Gong, "Web Structure Mining: An Introduction", Proceedings of the 2005 IEEE International Conference on Information Acquisition June 27 - July 3, 2005, Hong Kong and Macau, China.

[14] Ricardo Baeza-Yates and Emilio Davis ,"Web page ranking using link attributes" , In proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, PP.328-329, 2004.

[15] PRANAM KOLARI AND ANUPAM JOSHI, University of Maryland, Baltimore County, “WEB MINING: RESEARCH AND PRACTICE”, 2004 IEEE Copublished by the IEEE CS and the AIP.

[16] Deng Cai, Xiaofei He, Ji-Rong Wen, and Wei-Ying Ma, “Block-level Link Analysis”, In Proceedings of ACM SIGIR’04, July 25-29, 2004, Sheffield, South Yorshire, UK. 440-447.

[17] Taher H. Haveliwala, "Topic-Sensitive Page Rank: A Context-Sensitive Ranking Algorithms for Web Search", IEEE transactions on Knowledge and Data Engineering Vol.15, No 4 July/August 2003.

[18] T. Brants, F. Chen, and I. Tsochantaridis. “Topic-Based Document Segmentation with Probabilistic Latent Semantic Analysis”. In: Proceedings of CIKM’02, November 4-9,2002, McLean, Virginia, USA. 211-218.

[19] R. Kosala, H. Blockeel, “Web Mining Research: A Survey”, SIGKDD Explorations, Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining Vol. 2, No. 1 pp 1-15, 2000.

[20] A. Broder, R. Kumar, F Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, J. Wiener, “Graph Structure in the Web”, Computer Networks: The International Journal of Computer and telecommunications Networking, Vol. 33, Issue 1-6,pp309-320,2000.