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

Ashish Jain¹, Dr. Gireesh Dixit²

¹M.Tech., Madhav Proudyogiki Mahavidyalaya, Bhopal
²Prof. And Head(Comp. Sc.), Madhav Proudyogiki Mahavidyalaya, Bhopal

Abstract—The 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 PageRank (PR), Weighted PageRank (WPR), Hyperlink-Induced Topic Search (HITS) etc. The objective of this research is to discover an efficient and better system to identify authoritative web pages and to compare the results with existing algorithms. This new method is named as Intelligent Search Method (ISM).

Keywords—HITS, PageRank, WPR, Authoritative Web Pages, Web Structure Mining, Link Analysis, ISM.

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 pages 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 store 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. PAGE RANK

2.1 PageRank Algorithm

This algorithm was developed by Sergey Brin and Lawrence Page at Stanford University and is named after Lawrence Page.
PageRank is a link analysis algorithm which extends the idea of citation analysis. It considers 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 PageRank algorithm is based on web graph created by considering web pages as nodes and hyperlinks as edges. The rank value of a page is a measure of importance of that page. The numerical weight of a page A is denoted by PR(A) or Page Rank of page A.

We can define PageRank as follows:

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

\[ PR(A) = (1-d) + d \left( \frac{PR(T1)}{C(T1)} + \cdots + \frac{PR(Tn)}{C(Tn)} \right) \]

Here,

- \( PR(Ti) \) – is page rank of page Ti
- \( d \) – is the damping factor, \( 0 \leq d \leq 1 \) (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, at any step, that the person will continue. 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 PageRank of all the pages are 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:

\[ PR(A) = PR(B) + PR(C) + PR(D) + PR(E) \]

\[ = 0.2 + 0.2 + 0.2 + 0.2 \]

\[ = 0.8 \]

So PageRank 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 PageRank 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:

\[ PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{2} + PR(D) + \frac{PR(E)}{4} \]

\[ = 0.1 + 0.2 + 0.2 + 0.05 \]

\[ = 0.55 \]

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

\[ PR(A) = \frac{PR(B)}{C(B)} + \frac{PR(C)}{C(C)} + \frac{PR(D)}{C(D)} + \frac{PR(E)}{C(E)} \]

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

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

\[ PR(u) = \sum_{v \in B_u} \frac{PR(v)}{C(v)} \]

\( PR(u) \) is PageRank of page \( u \). PageRank 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 \( B_u \).

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 PageRank.

Figure 1 show how these Page Rank calculations are happening.

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

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

III. WEIGHTED PAGE RANK

This algorithm can be considered as an extension of PageRank 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 PageRank algorithm. As explained above, in PageRank 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 pages of the system will be having different weights. The weight of the page is decided in terms of incoming and outgoing links.

\[ W_{in}(m,n) = \frac{I_n}{\sum_{p \in R(m)} I_p} \]

In the above formula \( I_n \) is total number of incoming links of page \( n \) and \( I_p \) is total number of incoming links of page \( p \). The calculation of \( W_{in}(m,n) \) is done on the basis of the number 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 \( W_{in}(m,n) \):

\[ W_{out}(m,n) = \frac{O_n}{\sum_{p \in R(m)} O_p} \]

In the above formula \( O_n \) is total number of outgoing links of page \( n \), \( O_p \) is total number of outgoing links of page \( p \). So we can apply following formula to calculate \( W_{out}(m,n) \):

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 (R): 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 a page neither contain keywords of the query nor the relevant information then it is classified as IR (irrelevant Page).

We can compare PageRank 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 follows:

\[ k = \sum_{i \in R(p)} (n - i) \times W_i \]

Here \( i \) is the \( i \)th page in the result page-list \( R(p) \), \( n \) represents the first \( n \) pages chosen from the list \( R(p) \), and \( W_i \) is the weight value of \( i \)th page.

\[ W_i = (v1, v2, v3, v4) \]

Here, \( v1 \), \( v2 \), \( v3 \) and \( v4 \) are the values assigned to a page if the page is VR, R, WR and IR respectively. It is also obvious that
Experiments conducted on these algorithms prove that we get higher relevancy values with WPR as compared to Page Rank.

IV. 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 is possible that a page may be a good hub as well as a good authority at the same time.

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

\[ G = (V, E) \]

We can consider \( V \) is a set of vertices that ndian nt pages and \( E \) is set of edges that represent links.

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

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

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 referer pages of page \( p \), the authority weight of a page is proportional to the sum of hub weights of pages that link to it. Similarly a hub of a page is proportional to the sum of authority weights of pages that it links to.

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.

V. COMPARISON

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

1. Working of Algorithm: The PageRank calculates scores at the time of indexing and results are stored according to importance of pages. The Weighted PageRank also work in the same manner. The HITS computes Hub and Authority scores of n highly relevant pages on the fly.
2. Input Parameters: For PageRank input parameter is BackLinks Only. For Weighted PageRank input parameter is BackLinks and ForwardLinks. For HITS input parameter is BackLinks, ForwardLinks and Content.

3. Complexity:
   - PageRank – $O(\log N)$
   - Weighted PageRank - $< O(\log N)$
   - HITS - $< O(\log N)$

VI. 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 the “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.

VII. PROPOSED METHOD

The new method is named as “Intelligent Search Method (ISM)”. 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 results.

The method is tested by taking some sample queries. First the query is posted in is 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 ISM database (7.2). This database is a simple database containing a table called Interpret_Query. The table has two columns: Original_Query and Interpretation. The getInterpretation() method (7.3) takes Original query as parameter and it return interpretation of the query.

The Java implementation of the method is given in section 7.3. 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.

7.1 ISM Algorithm

Step 1: Input Search Query

Step 2: Generate interpretations of search queryion using getInterpretation() method and ISM Database given in the section 7.2 and 7.3.

Step 3: Post interpreted query to the search engine.

Step 4: generate the search results.

7.2 ISM Database

<table>
<thead>
<tr>
<th>Original_Query</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Narayan Murthy except Infosys Founder</td>
<td>Narayan Murthy – Infosys</td>
</tr>
<tr>
<td>XML tutorial on website w3schools.com</td>
<td>XML tutorial site:w3schools.com</td>
</tr>
<tr>
<td>sites similar to facebook.com</td>
<td>related: <a href="http://www.facebook.com">www.facebook.com</a></td>
</tr>
<tr>
<td>Only PDF tutorial on Database Management System</td>
<td>“Database Management System” tutorial filetype:pdf</td>
</tr>
<tr>
<td>books on C#.Net from 2002 to 2010</td>
<td>C#.Net books 2002..2010</td>
</tr>
<tr>
<td>useful links on DBMS</td>
<td>inanchor:DBMS</td>
</tr>
<tr>
<td>Url including the word company</td>
<td>inurl: company</td>
</tr>
<tr>
<td>Indian classical dance form except bharatnatyam</td>
<td>Indian classical dance form – bharatnatyam</td>
</tr>
</tbody>
</table>

7.3 Java implementation of getInterpretation() Method.

```java
class ISM {
    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;
}

VIII. EXPERIMENTS AND 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 contain number of relevant results out of first 30 results given by the google search engine.
3. Column 3 (Interpreted Query) : This column contain 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.
<table>
<thead>
<tr>
<th>Original Query</th>
<th>Page Rank</th>
<th>Interpreted query (Using ISM)</th>
<th>ISM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Narayan Murthy except Infosys Founder</td>
<td>2</td>
<td>Narayan Murthy –Infosys</td>
<td>9</td>
</tr>
<tr>
<td>XML tutorial on website w3schools.com</td>
<td>16</td>
<td>XML tutorial site:w3schools.com</td>
<td>30</td>
</tr>
<tr>
<td>sites similar to facebook.com</td>
<td>12</td>
<td>related: <a href="http://www.facebook.com">www.facebook.com</a></td>
<td>2</td>
</tr>
<tr>
<td>Only PDF tutorial on Database Management System</td>
<td>19</td>
<td>&quot;Database Management System&quot; tutorial filetype:pdf</td>
<td>24</td>
</tr>
<tr>
<td>books on C#.Net from 2002 to 2010</td>
<td>15</td>
<td>C#.Net books 2002..2010</td>
<td>27</td>
</tr>
<tr>
<td>useful links on DBMS</td>
<td>16</td>
<td>inanchor:DBMS</td>
<td>18</td>
</tr>
<tr>
<td>Url including the word company</td>
<td>4</td>
<td>inurl: company</td>
<td>4</td>
</tr>
<tr>
<td>indian classical dance form except bharatnatyam</td>
<td>5</td>
<td>Indian classical dance form –bharatnatyam</td>
<td>21</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>89</strong></td>
<td></td>
<td><strong>135</strong></td>
</tr>
</tbody>
</table>
The experimental results are also shown in following bar chart.

IX. CONCLUSION

It is clear from the above experiment that ISM 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 better and more relevant search results.

REFERENCES


[29] http://ptuce.loremate.com/dw/node/7


[38] http://ilpubs.stanford.edu:8090/422/


[41] Claudia Elena Dinucă, Craiova, Romania, “ON TWO ALGORITHMS USED IN WEB STRUCTURE MINING”.